

**DISEÑO Y CONSTRUCCIÓN DE UN DISPENSADOR AUTOMÁTICO DE  
ALIMENTO PARA EQUINOS**

**MARÍA FERNANDA HERNÁNDEZ RUEDA  
KILIAN ALEJANDRO RODRIGUEZ ROCHA**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE IGENIERIAS FISICO MECÁNICAS  
ESCUELA DE INGENIERÍA MECÁNNICA  
BUCARAMANGA**

**2021**

**DISEÑO Y CONSTRUCCIÓN DE UN DISPENSADOR AUTOMÁTICO DE  
ALIMENTO PARA EQUINOS**

**MARÍA FERNANDA HERNÁNDEZ RUEDA  
KILIAN ALEJANDRO RODRIGUEZ ROCHA**

**Trabajo de Grado para Optar el Título de:  
Ingeniero Mecánico**

**DIRECTOR  
Jorge Enrique Meneses Flórez  
MSc en Ingeniería Mecánica**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE IGENIERIAS FISICO MECÁNICAS  
ESCUELA DE INGENIERÍA MECÁNNICA  
BUCARAMANGA**

**2021**

## **DEDICATORIA**

A mis padres Luz Stella Rueda y José Hernández, quienes siempre me han guiado, me han apoyado de manera incondicional, confiaron en mí y me han enseñado que todo esfuerzo tiene su recompensa. Gracias por enseñarme a ser quien soy y este logro se los debo a ustedes.

A mis hermanos Karol Bibiana y Christian Iván por creer en mí, por brindarme su amor y su apoyo a pesar de la distancia y por ser un ejemplo para mí, los quiero.

**María Fernanda**

Dedico este y todos mis triunfos a mis padres, mi hermana, mi abuela y el resto de mi familia.

A todos los profesores con los que tuve clases y a las personas que me guiaron en el camino dedico este título universitario.

Dedico este triunfo a una persona muy importante, que siempre me apoyo, Doña Irma Núñez.

**Kilian Alejandro**

## **AGRADECIMIENTOS**

Primero quiere agradecerle a Dios porque en cada paso pude ver que estuvo acompañándome.

A mis padres Luz Stella y Jose Hernandez quienes siempre me han demostrado su apoyo incondicional, me enseñaron que, aunque se tengan dificultades para alcanzar una meta nunca debo dejar de luchar por ello, gracias por sus esfuerzos, su dedicación y por cada sacrificio, cada logro que he tenido y que llegue a tener es de ustedes.

A mis hermanos Christian Iván y Karol Bibiana por su apoyo, por sus consejos, son tan increíbles y únicos, lo mejor que la vida me ha dado, sin duda mi mejor ejemplo y el reflejo de lo afortunada que soy.

A Juan Jose por estar durante mis crisis al momento de realizar este trabajo, por levantarme el ánimo y ser mi gran animador, gracias por permitirme conocerte y estar en mi vida.

A mis amigos gracias. Junior porque con su forma de ser me alegraba el día con cada palabra a pesar de la distancia. Camila, gracias por seguirme la corriente en tantas cosas y ser mi hermana de corazón, los quiero.

**María Fernanda**

## **AGRADECIMIENTOS**

Agradecer primero que todo a Dios, porque sin él no hubiera podido lograr nada en la vida, ni estaría donde estoy.

Agradezco de manera especial a mis padres Pedro Jesús Rodríguez Gaitán y Reina Amparo Rocha Rocha porque gracias a su gran esfuerzo para brindarme el apoyo moral y económico estoy cumpliendo mis sueños; Para mi padre solo palabras de admiración pues siendo una persona con estudios básicos de primaria ha llegado muy lejos y me ha demostrado que con esfuerzo se pueden lograr grandes cosas; A mi madre le debo el estar estudiando y aunque no fui fácil, ella nunca se rindió siempre creyó en mi cuando nadie más lo hacía, la mejor madre que una persona pudiera tener.

Aunque el camino para llegar donde estoy no ha sido fácil, con esfuerzo, dedicación y la ayuda de muchas personas he podido avanzar hasta lograr conseguir el título de ingeniero mecánico UIS, agradezco a todas las personas que he conocido pues de todas he aprendido algo.

Quiero expresar también mis sinceros agradecimientos a Iván Mauricio Castellanos estudiante de ingeniería electrónica por su apoyo, conocimiento y paciencia, la mayoría de lo que se dé electrónica se lo debo a él, de corazón le agradezco seguirme la corriente en todos los proyectos que se me ocurren.

También quiero agradecerle al ingeniero electrónico Sergio Moreno por su apoyo al ayudarme a solucionar y guiarme en cosas que no sabía cómo hacerlas.

Expresar mi gran agradecimiento para con la Universidad Industrial de Santander, sus profesores, sus administrativos, celadores, las personas de aseo y a todas las personas que hacen de la universidad lo que es.

**Alejandro**

## CONTENIDO

	Pág.
INTRODUCCIÓN .....	15
1.OBJETIVOS.....	17
1.1. OBJETIVO GENERAL .....	17
1.2. OBJETIVOS ESPECÍFICOS.....	17
2.DESCRIPCIÓN DEL DISEÑO Y CONSTRUCCIÓN DE UN DISPENSADOR AUTOMÁTICO DE ALIMENTO PARA EQUINOS.....	19
2.1. SISTEMA MECÁNICO .....	20
2.1.1. Subsistema de almacenamiento .....	20
2.1.2. Subsistema de distribución .....	21
2.2. HARDWARE .....	24
2.2.1. Subsistema de Control.....	24
2.2.2. Subsistema Operacional.....	25
2.2.3. Subsistema energético.....	25
2.3. SOFTWARE.....	25
2.3.1. Software microcontrolador. ....	25
2.3.2. Software aplicación Android.....	25
2.4. ANÁLISIS DATOS DE DOSIFICACIÓN Y RESULTADOS .....	26
2.5. MANUAL DE USUARIO.....	26
3.DISEÑO MECÁNICO DEL SISTEMA .....	27
3.1. DISEÑO SUBSISTEMA DE ALMACENAMIENTO.....	27
3.1.1. Diseño del subsistema de distribución.....	30
3.1.2. Alternativas de diseño del sistema mecánico. ....	31
3.2. ANÁLISIS POR ELEMENTOS FINITOS .....	38
4.DISEÑO HARDWARE .....	42
4.1. SUBSISTEMA DE CONTROL .....	42
4.1.1. Arduino MEGA 2560 R3.....	42
4.1.2. Modulo conversor I2C.....	43
4.1.3. Pantalla LCD 20x4. ....	44
4.1.4. Teclado matricial 4x4 formato PCB 16 teclas. ....	45
4.1.5. Reloj DS3231.....	46
4.1.6. Módulo Bluetooth HC-05.....	47
4.1.7. Sensor ultrasonico HC-SR04.....	48
4.2. SUBSISTEMA OPERACIONAL.....	48

4.2.1. Servomotor SG90. ....	48
4.2.2. Servomotor MG996R. ....	49
4.3. SUBSISTEMA ENERGÉTICO .....	50
4.3.1. Adaptador Corriente Alterna (AC) / Corriente Directa (DC).....	50
4.4. DIAGRAMA ESQUEMÁTICO MODULO DE CONTROL .....	50
5.DISEÑO SOFTWARE .....	52
5.1. SOFTWARE DE PROGRAMACIÓN .....	52
5.1.1. Descripción general .....	52
5.1.2. Reloj DS3231 .....	54
5.1.3. Alarma.....	55
5.1.4. Pantalla LCD y teclado matricial. ....	56
5.1.5. Comunicación Bluetooth. ....	57
5.1.6. Programación raciones. ....	58
5.1.7. Calibración celdas de carga.....	62
5.1.8. Conservación de datos. ....	64
5.2. INTERFAZ HUMANO – MÁQUINA .....	65
5.2.1. IHM Local.....	65
5.2.2. IHM Remota.....	71
6.PRUEBAS Y RESULTADOS .....	78
6.1. PROGRAMACIÓN LOCAL PARA DISPENSAR EL ALIMENTO A UN EQUINO .....	78
7.MANUAL DEL USUARIO .....	83
8.COSTOS DEL EQUIPO .....	84
9.CONCLUSIONES Y RECOMENDACIONES .....	85
BIBLIOGRAFIA .....	86

## LISTA DE TABLAS

	<b>Pág.</b>
Tabla 1. Muestras de peso de alimento en volumen específico. ....	29
Tabla 2. Datos estadísticos de las muestras .....	29
Tabla 3. Densidad de las muestras.....	30
Tabla 4. Listado de elementos usados para el Hardware .....	42
Tabla 5. Cantidad de alimento entregado al equino 1.....	78
Tabla 6. Cantidad de alimento entregado al equino 1.....	79
Tabla 7. Alimento entregado al equino 2. ....	80
Tabla 8. Cantidad de alimento entregado al equino 2.....	81
Tabla 9. Costos del equipo. ....	84

## LISTA DE FIGURAS

	<b>Pág.</b>
Figura 1. Subsistemas del distribuidor automático de alimento .....	19
Figura 2. Subsistema de almacenamiento .....	210
Figura 3. Subsistema de distribución .....	22
Figura 4. Sistema de automatización .....	23
Figura 5. Microcontrolador subsistema de control .....	24
Figura 6. Ángulo de reposo estático. ....	28
Figura 7. Alternativa 1 diseño del equipo. ....	32
Figura 8. Alternativa 2 diseño del equipo. ....	34
Figura 9. Alternativa 3 diseño del equipo .....	36
Figura 10. Tensiones análisis estático. ....	38
Figura 11. Desplazamiento análisis estático. ....	39
Figura 12. Deformaciones análisis estático. ....	40
Figura 13. Factor de seguridad análisis estático. ....	41
Figura 14. Pinout Arduino Mega 2560 R3. ....	43
Figura 15. Pinout modulo conversor I2C. ....	44
Figura 16. Pantalla LCD 20x4. ....	45
Figura 17. Teclado matricial 4x4. ....	45
Figura 18. Pinout reloj DS3231. ....	46
Figura 19. Pinout módulo HC05. ....	47
Figura 20. Pinout sensor ultrasónico HC-SR04. ....	48
Figura 21. Conexiones servomotor MG996R. ....	49
Figura 22. Diagrama esquemático .....	51
Figura 23. Pantalla inicial. ....	53
Figura 24. Void setup y void loop .....	53
Figura 25. Carpetas usadas. ....	54

Figura 26. Programación para el control del reloj .....	55
Figura 27. Generación de alarmas.....	55
Figura 28. Teclado matricial.....	56
Figura 29. Vectores.....	56
Figura 30. Creación de pantallas .....	57
Figura 31. Variables de comunicación .....	57
Figura 32. Conversión de vectores para pantalla LCD. ....	58
Figura 33. Programación de raciones .....	60
Figura 34. Calibración de celdas de carga.....	63
Figura 35. Conservación de datos .....	64
Figura 36. IHM Local. Hora y fecha .....	66
Figura 37. IHM Local. Modo manual.....	66
Figura 38. IHM Local. Modificación de valores .....	67
Figura 39. IHM Local. Modificación de valores, caballo 1. ....	67
Figura 40. IHM Local. Modificación de valores, caballo 2. ....	68
Figura 41. IHM Local. Opciones de preferencia.....	68
Figura 42. IHM Local. Ejecutando.....	69
Figura 43. IHM Local. Pantalla principal .....	69
Figura 44. Diagrama de flujo pantallas en LCD. ....	70
Figura 45. IHM Remota. Interfaz.....	71
Figura 46. IHM Remota.Programación. ....	72
Figura 47. IHM Remota. Funciones principales. ....	73
Figura 48. IHM Remota.Interfaz de ingreso. ....	75
Figura 49. IHM Remota. Ingreso.....	75
Figura 50. IHM Remota. Configuración de la cantidad de alimento .....	76
Figura 51. Cambio en la pantalla de modo bluetooth.....	76
Figura 52. Selección de suministro de alimento.....	77

## LISTA DE FOTOGRAFÍAS

	<b>Pág.</b>
Fotografía 1. Balanza y recipiente de pesaje .....	28
Fotografía 2. Prototipo de la alternativa 1. ....	33
Fotografía 3. Prototipo de la alternativa 2. ....	35
Fotografía 4. Prototipo final alternativa seleccionada. ....	37

## **LISTA DE ANEXOS**

**(Los anexos están adjuntos y puede visualizarlos en base de datos de la biblioteca UIS)**

Anexo A. Fotografías toma peso de las muestras.

Anexo B. Planos mecánicos del equipo.

Anexo C. Marco Teórico.

Anexo D. Marco Legal.

Anexo E. Prototipo alternativa 1 de diseño del equipo.

Anexo F. Prototipo alternativa 2 de diseño del equipo.

Anexo G. Prototipo alternativa 3 de diseño del equipo.

Anexo H. Manual del usuario

Anexo I. Videos pruebas al dispensador

## RESUMEN

**TÍTULO:** DISEÑO Y CONSTRUCCIÓN DE UN DISPENSADOR AUTOMÁTICO DE ALIMENTO PARA EQUINOS\*

**AUTORES:** MARÍA FERNANDA HERNÁNDEZ RUEDA\*\*  
KILIAN ALEJANDRO RODRIGUEZ ROCHA\*\*\*

**PALABRAS CLAVE:** DISPENSADOR AUTOMÁTICO, ARDUINO, EQUINOS, APLICACIÓN MÓVIL, CELDAS DE CARGA, BLUETOOTH.

### DESCRIPCIÓN:

El ser humano ha utilizado a los equinos en áreas como la salud, la guerra y la agricultura durante el desarrollo y crecimiento de la sociedad para beneficio propio. Sin embargo, su domesticación trae retos debido al cuidado que el dueño ha de tener con el equino en cuanto a sus hábitos alimenticios, un mal manejo del mismo conlleva a serios trastornos de alimentación en el animal, el principal es conocido como cólico equino y es especialmente consecuencia de la frecuencia y cantidad de alimento con la cual es alimentado.

Teniendo en cuenta lo anterior, este proyecto se basó en diseñar un equipo mecatrónico que contribuya a suplir la necesidad del sector ecuestre relacionadas con el cuidado apropiado de los animales, proporcionando las cantidades adecuadas de alimento con la frecuencia necesaria que tiene el equino para asemejar su dieta a la del hábitat natural.

Enfocados en garantizar la cantidad adecuada de alimento entregado al equino de manera independiente al tipo de alimento utilizado se usaron celdas de carga y para el control del dispensador automático se utilizó una programación basada en Arduino. El usuario tiene la opción de configurarlo mediante un teclado matricial ubicado en el equipo, además, cuenta con la capacidad de ser programado por medio de una aplicación móvil en su Smartphone Android, a través de una conexión bluetooth lo que facilitará el cuidado del mismo.

---

\* Trabajo de Grado

\*\* Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería Mecánica Director Jorge Enrique Meneses Flórez.

## ABSTRACT

**TITLE:** DISIGNED AND CONSTRUCTION OF AN AUTOMATIC FEEDER FOR HORSES\*

**AUTHORS:** MARÍA FERNANDA HERNÁNDEZ RUEDA\*\*  
KILIAN ALEJANDRO RODRIGUEZ ROCHA\*\*

**KEY WORDS:** AUTOMATIC FEEDER, ARDUINO, HORSE, MOBILE APLLICATION, LOAD CELL, BLUETOOTH.

### DESCRIPTION:

Human beings have used equines in areas such as health, war and agriculture during the development and growth of society for their own benefit. However, their domestication brings challenges due to the care that the owner must take with the equine in terms of its feeding habits, its reduced management leads to serious eating disorders in the animal, the main one is known as equine colic and is especially a consequence of the frequency and quantity of food with which it is fed.

Taking into account the above, this project was based on designing a mechatronic equipment that contributes to meet the need of the equestrian sector related to the proper care of animals, providing adequate amounts of food with the necessary frequency that the equine has to resemble their diet to that of the natural habitat.

Focused on guaranteeing the proper amount of feed delivered to the equine, regardless of the type of food, load cells were used, it uses a programming based on Arduino to control the equipment. The user has the option to configure it using a matrix keyboard located on the equipment, also has the ability to be programmed using a mobile application on your Android Smartphone, through a bluetooth connection which will facilitate the care of it.

---

\* Degree project

\*\*Faculty of Engineering Physics and Mechanics. School of Mechanical. Head Teacher Jorge Enrique Meneses Flórez

## INTRODUCCIÓN

A través de la historia, los caballos, han sido de gran beneficio para el ser humano en diferentes áreas como lo son la guerra, la salud humana y la agricultura. En la actualidad los caballos son considerados como una mascota doméstica, esto implica que el animal debe tener cuidados apropiados.

Dentro de los cuidados apropiados se encuentran revisiones veterinarias, espacios adecuados, acceso al agua y una dieta balanceada. Es en este último en el cual no se ha logrado satisfacer la necesidad del animal por completo. Esto debido a que en la naturaleza el tipo de alimentación de los caballos es constante, pero en pequeñas porciones así que al domesticar los equinos la forma de alimentarlos cambió, lo cual puede provocar problemas de salud en el caballo.

El sistema digestivo de los caballos es delicado debido a sus condiciones anatómicas, esto los hace propensos a sufrir del síndrome de abdomen agudo en el equino, más conocido como, cólico, algunos de los síntomas que puede presentar son revolcarse repetidamente, dar patadas, negarse a comer o a beber agua. La principal razón para sufrir de este síndrome es la calidad y la cantidad de la ingesta de alimento, bien sea una dieta excesiva o escasa.

Actualmente el método más utilizado para alimentar a los equinos es de manera manual, es decir, un empleado u operario se encarga de llevar el alimento del caballo a determinadas horas, pero no en las porciones y en las cantidades adecuadas para el mismo. En consecuencia, estos factores afectan notablemente en relación con los gastos operativos derivados de una incorrecta distribución del alimento generando más desperdicios, así también, se presentan gastos veterinarios debido a las enfermedades gastrointestinales ocasionadas por esta mala práctica.

El desarrollo de nuevas tecnologías incorporadas al entorno de los equinos permitirá mejorar los métodos productivos, pues el proceso de alimentación manual llevado a cabo por los operarios no es muy eficaz.

En el proyecto se desarrolló un sistema de dosificación automático de alimento para los equinos con la finalidad de aportar una solución al problema actual, este provee al usuario la posibilidad de controlar las siguientes características:

- Calibración y control del sistema mediante una interfaz hombre-máquina con el uso de una aplicación móvil.
- Correcta dosificación de alimento de los equinos, de tal manera que se cumplan los requerimientos alimenticios del animal, repartiendo eficientemente las raciones, esto con el fin de reducir efectos en la salud del animal generados por malos manejos en su alimentación.

De esta manera se logra optimizar el proceso de alimentación, pues se está dando un mayor aprovechamiento de la comida permitiendo al cliente mejorar la crianza de los caballos, así mismo se contribuye a problemáticas en términos de tiempo y operación, lo que infiere menos gastos, se estaría eliminando en su mayoría el trabajo del operario.

# 1. OBJETIVOS

## 1.1. OBJETIVO GENERAL

Fortalecer la relación del trabajo académico con el mundo externo, en concordancia con la misión de la Universidad Industrial de Santander, implementando un equipo mecatrónico que contribuya con las necesidades relacionadas al sector ecuestre para atenderlas de forma sostenible.

## 1.2. OBJETIVOS ESPECÍFICOS

1. Diseñar y construir un equipo mecatrónico de dosificación de alimento concentrado para equinos que cumpla con las siguientes características:
  - Poseer una capacidad de 16 Kg para ser dispensados durante dos (2) días a dos (2) equinos.
  - Automatización a nivel hardware
    - ✓ Programación basada en Arduino para el control del equipo.
    - ✓ Una interfaz (HMI) de programación, calibración y seguimiento basada en smartphone con sistema Android y conexión bluetooth.
    - ✓ Una interfaz (HMI) local para visualización de la programación de dosificación en ejecución, las alarmas generadas, la fecha y hora absolutas de operación del equipo.
    - ✓ Retención de datos semipermanente (2 días), en ausencia del suministro eléctrico al equipo.
  - Automatización a nivel software

- ✓ Dosificar la alimentación, en el tiempo, configurable para cada equino
- ✓ Calibrar la dosificación del equipo
- ✓ La identificación de los estados de parada del equipo (funcionamiento anormal) y generación de las respectivas alarmas
- ✓ El almacenamiento de los datos relevantes de operación, para la supervisión del equipo
- ✓ Niveles de seguridad que garanticen el acceso de usuarios autorizados.

2. Elaborar el manual de operación y mantenimiento que facilite el uso del equipo.

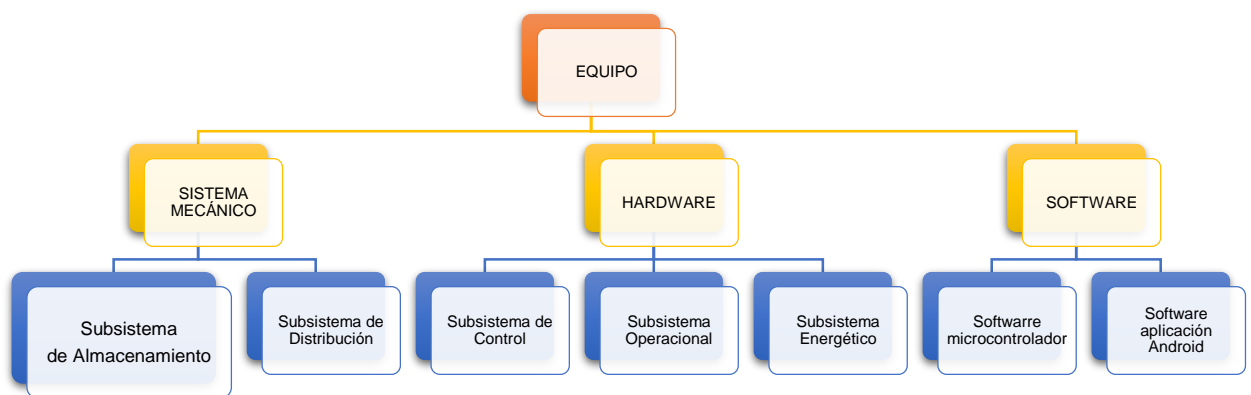
## 2. DESCRIPCIÓN DEL DISEÑO Y CONSTRUCCIÓN DE UN DISPENSADOR AUTOMÁTICO DE ALIMENTO PARA EQUINOS

De acuerdo a las necesidades especiales para alimentar un equino, los dueños de este tipo de animales han encontrado dificultades en la manera adecuada de ejecutar esta función, por lo general, lo hacen de manera manual esto implica que un operario va a determinadas horas y se encarga de llenar el plato del animal, sin embargo, es ineficiente en aspectos como los horarios y las cantidades recomendadas para el equino.

Como segunda opción tienen un dispensador automático, este se encarga de entregar el alimento en un horario establecido y la cantidad requerida, a pesar de existir este tipo de equipos en el mercado se encuentra que son capaces de cumplir su función en cuanto a los horarios, pero no cumplen en las cantidades de alimento entregadas al equino.

Para iniciar el diseño del equipo en el proyecto se dividió en diferentes subsistemas como se representa en la siguiente figura.

Figura 1. Subsistemas del distribuidor automático de alimento



Fuente: Autores

## **2.1. SISTEMA MECÁNICO**

**2.1.1. Subsistema de almacenamiento.** Para determinar las medidas se calculó la densidad de tres tipos de alimento con diferente granulometría y se eligió la de menor valor, de esta forma se garantiza que el equipo cumplirá con la capacidad de almacenamiento requerida.

Para su construcción se determinó que de acuerdo a la normatividad legal vigente el material idóneo es acero inoxidable 304 debido a que evita la contaminación del alimento y no desprende partículas tóxicas al momento de realizar la limpieza. Respecto a la elaboración del prototipo el material utilizado fue cartón industrial.

En cuanto a manufactura se determinó que por costos la mejor manera de fabricarlo era realizar un proceso de doblado (el desarrollo y dobleces requeridos se pueden ver en el anexo B donde están dos planos mecánicos) y se une mediante conexiones atornilladas para construir el cuerpo y la tolva en lugar de unir las piezas por medio de soldadura, esto disminuye los costos y facilita el mantenimiento del equipo al permitir desmontarlo de una manera rápida y económica.

Figura 2. Subsistema de almacenamiento



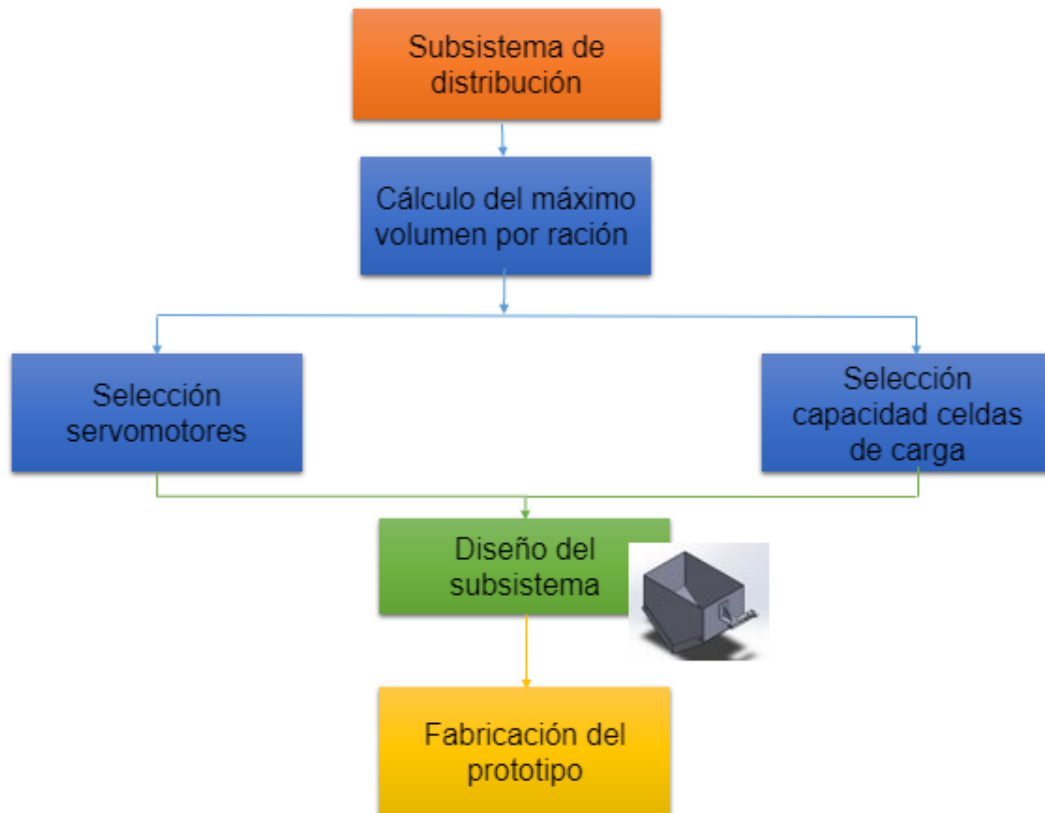
Fuente: Autores

**2.1.2. Subsistema de distribución.** Como se había mencionado antes se debía cumplir con el objetivo de entregar la cantidad adecuada de alimento para el equino, por lo que se estableció que la mejor opción era utilizar un dispensador por gravedad tipo báscula, para esto se utilizaron dos celdas de carga tipo viga, las cuales tienen la capacidad de 1 kg cada una, además, poseen la capacidad de ser calibradas y no se dependerá de la granulometría del alimento para el mismo.

El equipo entregará en una ración máximo 1 kg de pienso para los equinos. Para entregar el alimento a un caballo o al otro el recipiente que lo contiene mientras se desarrolla el proceso de pesaje cuenta con dos compuertas de salida, una ubicada para abrirse hacia el lado derecho y otra hacia el lado izquierdo, ellas son

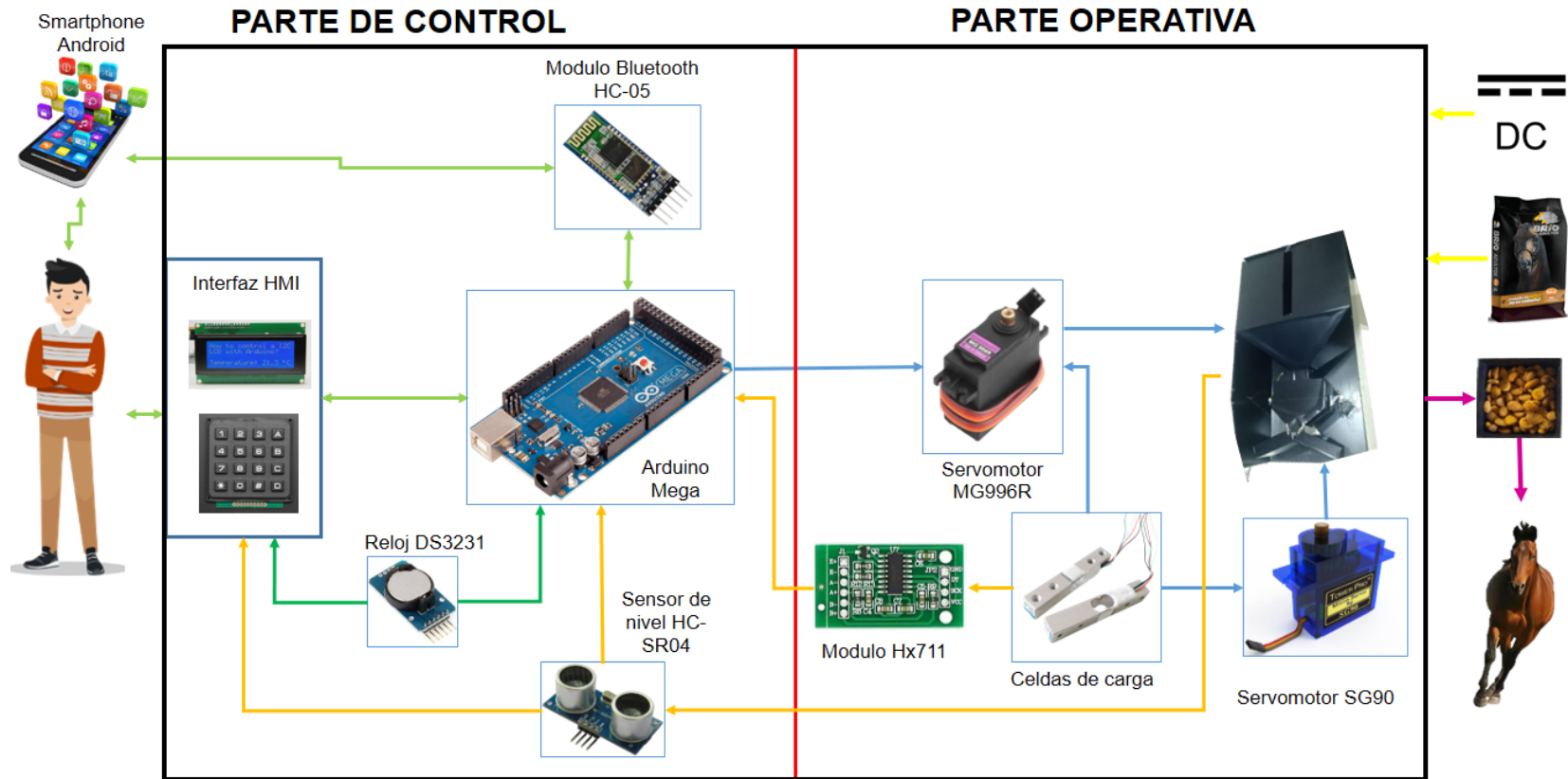
controladas a través de dos servomotores SG90, los cuales se activaran dependiendo del animal alimentar.

Figura 3. Subsistema de distribución



Fuente: Autores

Figura 4. Sistema de automatización

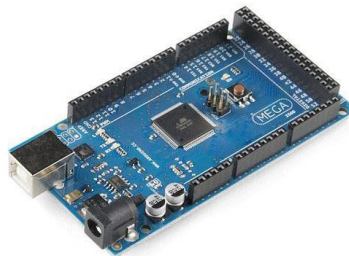


Fuente: Autores

## 2.2. HARDWARE

**2.2.1. Subsistema de Control.** Para el control del equipo se decidió utilizar una placa con microcontrolador Arduino mega 2560 R3 debido a que es de bajo costo, fácil de programar, cuenta con un software libre y extensible mediante una amplia gama de librerías. Tiene una alta capacidad de memoria lo que evita posibles errores de funcionamiento.

Figura 5. Microcontrolador subsistema de control



Fuente: GEEK FACTORY. Tarjeta genérica compatible con Arduino Mega 2560. 2013-2020. [En línea]. [Consultado el 12 de noviembre de 2020]. Disponible en: <https://www.geekfactory.mx/tienda/compatibles-con-arduino/arduino-mega-2560-generico/>

El equipo cuenta con una pantalla de cristal líquido LCD 20x4 para permitir la visualización de la hora, fecha, alarmas y cantidades programadas. La pantalla LCD se encuentra conectada a un módulo I2C para reducir la cantidad de pines utilizados en el Arduino.

Para la generación de la alarma de nivel del alimento almacenado se usó un sensor de nivel HC-SR04. La comunicación entre el equipo y la aplicación móvil se realiza a través del módulo bluetooth HC-05.

De manera adicional el dispensador tiene un teclado matricial de 4x4 de 16 teclas para programar la cantidad y los horarios de alimento de los equinos, esto para configurar el equipo en caso de no tener el celular disponible. Al igual que la pantalla

se encuentra conectada a un módulo I2C para reducir la cantidad de pines utilizados en el Arduino.

**2.2.2. Subsistema Operacional.** El equipo cuenta con un servomotor MG996R encargado de entregar el alimento desde el subsistema de almacenamiento hacia el subsistema de distribución.

En el subsistema de distribución el equipo dos celdas de carga de capacidad de 1 kg son las encargadas de sensar la cantidad de alimento que será entregado al caballo, quienes a su vez se encuentran conectadas a dos módulos Hx711, los cuales se encargan de transmitir la señal recibida por las celdas al Arduino.

Una vez se alcanza el peso programado en la ración, se activa uno de los dos servomotores SG90 dependiendo del caballo que se vaya a alimentar.

**2.2.3. Subsistema energético.** El método seleccionado para mantener energizados los componentes eléctricos y que de esta manera estén siempre listos para ser accionados fue conectar el equipo a una fuente eléctrica, debido al bajo costo de implementación y a la reducción en el peso del equipo.

## **2.3. SOFTWARE**

**2.3.1. Software microcontrolador.** El software utilizado para la programación del microcontrolador es Arduino. Se estableció como el manejado por la compañía fabricante de la placa debido a que es libre y tiene variedad de librerías.

**2.3.2. Software aplicación Android.** Para el desarrollo de la aplicación se decidió usar Android Studio. Este es de manera oficial el software para la creación de aplicaciones en sistema Android.

## **2.4. ANÁLISIS DATOS DE DOSIFICACIÓN Y RESULTADOS**

Una vez se fabricó el prototipo se procedió a realizar las pruebas con alimento para equinos, esto con la finalidad de evaluar el desempeño y el funcionamiento del dispensador.

## **2.5. MANUAL DE USUARIO**

Con el propósito de facilitar el manejo del equipo para la programación y mantenimiento, se elaboró un documento en el cual el usuario puede guiarse para saber el manejo adecuado del dispensador desde la aplicación Android o desde el teclado.

### 3. DISEÑO MECÁNICO DEL SISTEMA

#### 3.1. DISEÑO SUBSISTEMA DE ALMACENAMIENTO

Para el diseño del tanque de almacenamiento primero se determinó la cantidad de alimento que consumen dos caballos con peso promedio de 450 kg durante dos días, con una condición de trabajo intensa (1% de acuerdo a lo recomendado por la línea brío adultos).

$$450 \text{ kg} * 0.01 = 4.5 \text{ kg}$$

Con base en esto se determinó que la cantidad entregada por el dispensador en un día por cada caballo será de 4.5 kg. Entonces dos caballos durante dos días consumirían aproximadamente 18 kg de alimento.

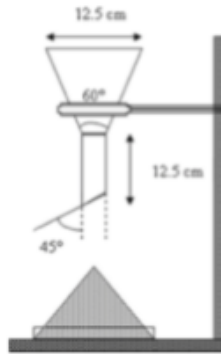
El diseño del subsistema de almacenamiento debe cumplir con los siguientes requisitos:

- Capacidad de almacenamiento de 18 kg.
- La unidad de almacenamiento debe ser resistente a deformaciones cuando se encuentre en su máxima capacidad.
- El material no debe afectar las características físicas o nutricionales del alimento.
- Tiene que ser práctica para su llenado.
- Accesos para limpieza y mantenimiento.
- No debe permitir el atasco de alimento.

Para generar una caída de alimento continua en la salida del subsistema de almacenamiento, se tuvo en cuenta el ángulo de reposo, está definido como el ángulo de la pendiente formada por el cono producido respecto al plano horizontal

cuando se deja caer libremente un material desde un embudo de vidrio. Entre menor sea el ángulo de reposo, mayor será el flujo del material y viceversa<sup>1</sup>.

Figura 6. Ángulo de reposo estático.



Fuente: HINOJOSA, Diego Mauricio. Diseño de dispensador automático de alimento para animales. Quito.: Universidad Tecnológica Equinoccial.

Es necesario determinar la densidad aparente del alimento para establecer el tamaño del tanque de almacenamiento, para esto se pesó el producto en un recipiente de volumen fijo.

Fotografía 1. Balanza y recipiente de pesaje



Fuente: Autores

---

<sup>1</sup> HINOJOSA, Diego Mauricio. Diseño de dispensador automático de alimento para animales. Quito.: Universidad Tecnológica Equinoccial.

Se pesó en repetidas ocasiones la cantidad contenida en el recipiente con cada una de las muestras de alimento (ver Anexo A), con ellos se calculó el peso medio en el recipiente.

Tabla 1. Muestras de peso de alimento en volumen específico.

Alimento	Medición									
	1	2	3	4	5	6	7	8	9	10
Corcel	78	78	77	78	78	77	79	77	78	79
Cinta azul	67	66	66	67	66	68	68	67	67	66
Brío	90	91	92	92	91	90	91	92	91	91

Fuente: Autores

Para el cálculo del peso medio en el recipiente de volumen fijo se usó la fórmula:

$$\bar{x} = \frac{N_1 + N_2 + N_3 + \dots + N_{10}}{N_T}$$

En la tabla 2, se muestran los datos estadísticos encontrados.

Tabla 2. Datos estadísticos de las muestras

ESTADÍSTICA DE MUESTRA				
Alimento	No. de muestras	Media	Valor mínimo	Valor máximo
Corcel	10	78	77	79
Cinta azul	10	67	66	68
Brío	10	91	90	92

Fuente: Autores

Se calculó la densidad aparente del alimento con la siguiente ecuación.

$$\rho = \frac{m}{v}$$

Donde  $\rho$  es la densidad,  $m$  es la masa y  $v$  es el volumen del recipiente. Los datos obtenidos se encuentran en la tabla 3.

Tabla 3. Densidad de las muestras

Marca	Peso	Densidad
Corcel	78	0,28
Cinta azul	67	0,49
Brío	91	0,51

Fuente: Autores

De acuerdo con lo anterior se tomó la densidad más crítica (alimento marca Corcel) de alimento para calcular el volumen que debe contener la tolva, esto se determinó con la fórmula:

$$V_{tolva} = \frac{18 \text{ kg}}{\rho_{critica}} = \frac{18000 \text{ g}}{0.28 \frac{\text{g}}{\text{cm}^3}} = 65077 \text{ cm}^3$$

Con el volumen hallado es posible encontrar las dimensiones adecuadas para el subsistema del equipo.

**3.1.1. Diseño del subsistema de distribución.** De acuerdo con los tipos de alimentos tenidos en cuenta como referencia lo recomendado por los veterinarios para una alimentación adecuada del equino, la cantidad máxima por porción en promedio que debe ser suministrada es de 1 kg.

Acorde con lo anterior, se usó el mismo método para determinar el volumen de la tolva y se estableció el volumen del recipiente que va a contener el alimento durante el proceso de pesaje y distribución.

$$V_{recipiente} = \frac{1 \text{ kg}}{\rho_{critica}} = \frac{1000 \text{ g}}{0.28 \frac{\text{g}}{\text{cm}^3}} = 3571 \text{ cm}^3$$

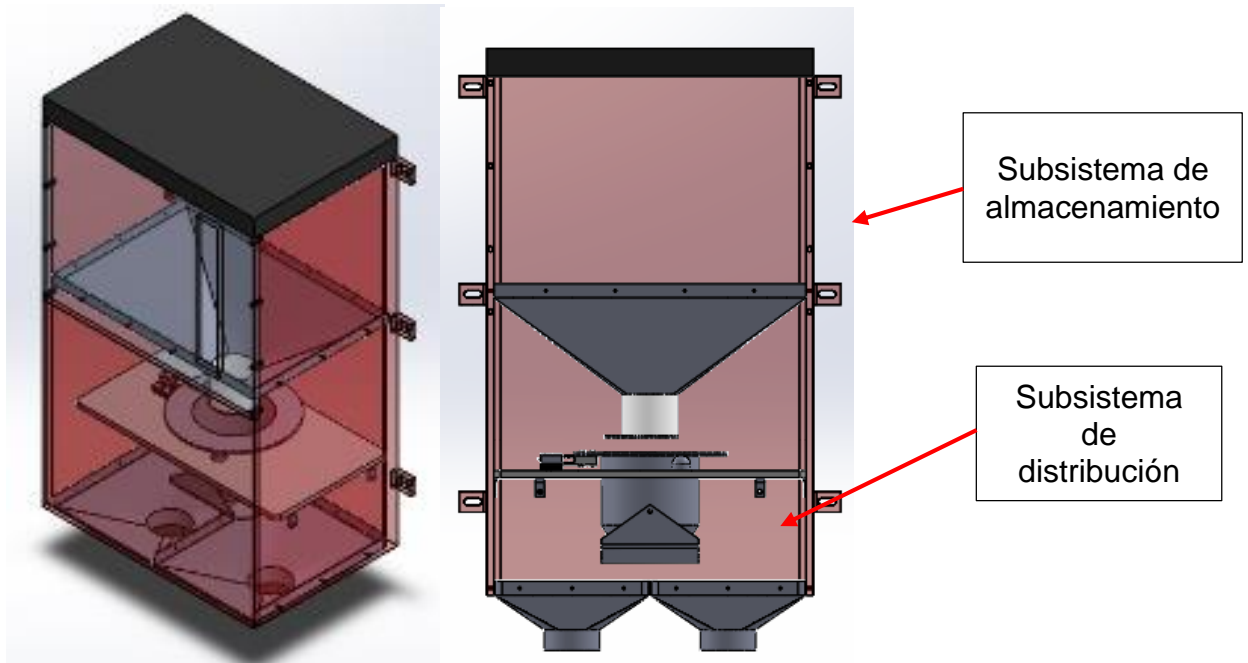
Con el volumen calculado es posible definir las dimensiones que debe tener el recipiente de pesaje del subsistema de distribución para cumplir con la ración máxima a entregar.

**3.1.2. Alternativas de diseño del sistema mecánico.** Primero se seleccionó el material para la fabricación, el cual se determinó en base a la normativa colombiana (ver anexo D) para la manipulación de alimentos de consumo animal, se seleccionó acero inoxidable 304 calibre no. 18. Para el diseño del equipo se usó como apoyo el software CAD Solid Works, en este se crearon alternativas al diseño del equipo.

Antes de fabricar el equipo se construyeron prototipos de las alternativas de diseño con cartón industrial, con ellos se estudiaron factores como la manufactura, funcionalidad y eficacia en la entrega de alimento.

### 3.1.2.1. Alternativa 1

Figura 7. Alternativa 1 diseño del equipo.

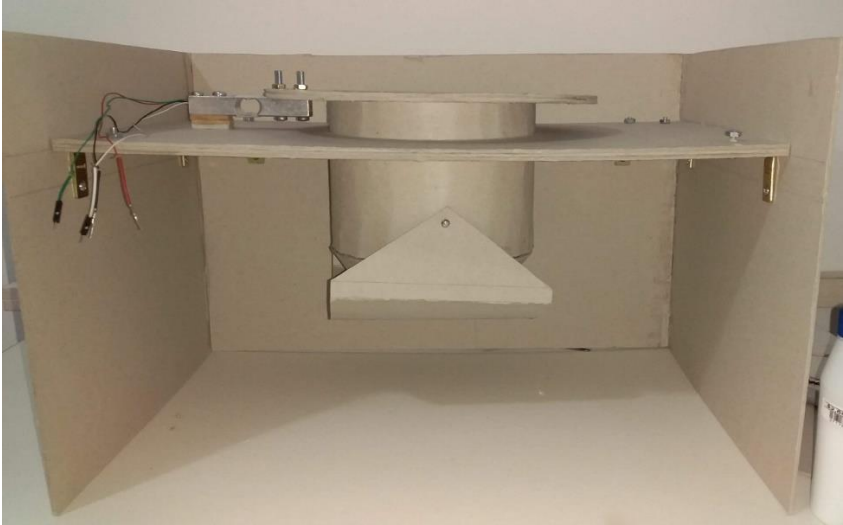


Fuente: Autores

En esta alternativa se plantea un subsistema de almacenamiento en forma prismática rectangular donde el subsistema de distribución es un recipiente en forma cilíndrica con un ala al borde la cual se une a la celda de carga. Para entregar el alimento se usa una lamina doblada y se encarga de balancearse hacia el lado izquierdo o derecho con un servomotor de acuerdo al caballo a alimentar, la forma de dirigir el alimento al plato adecuado es mediante dos tolvas al final del equipo.

No fue seleccionada debido a que se fabrico un prototipo del subsistema de distribución (Ver anexo E) donde se observó que presenta inconvenientes en la confiabilidad del peso sensado por la celda de carga e inestabilidad al sostener el recipiente de pesaje, adicionalmente desde el punto de vista economico eleva los costos de manufactura y material por su forma.

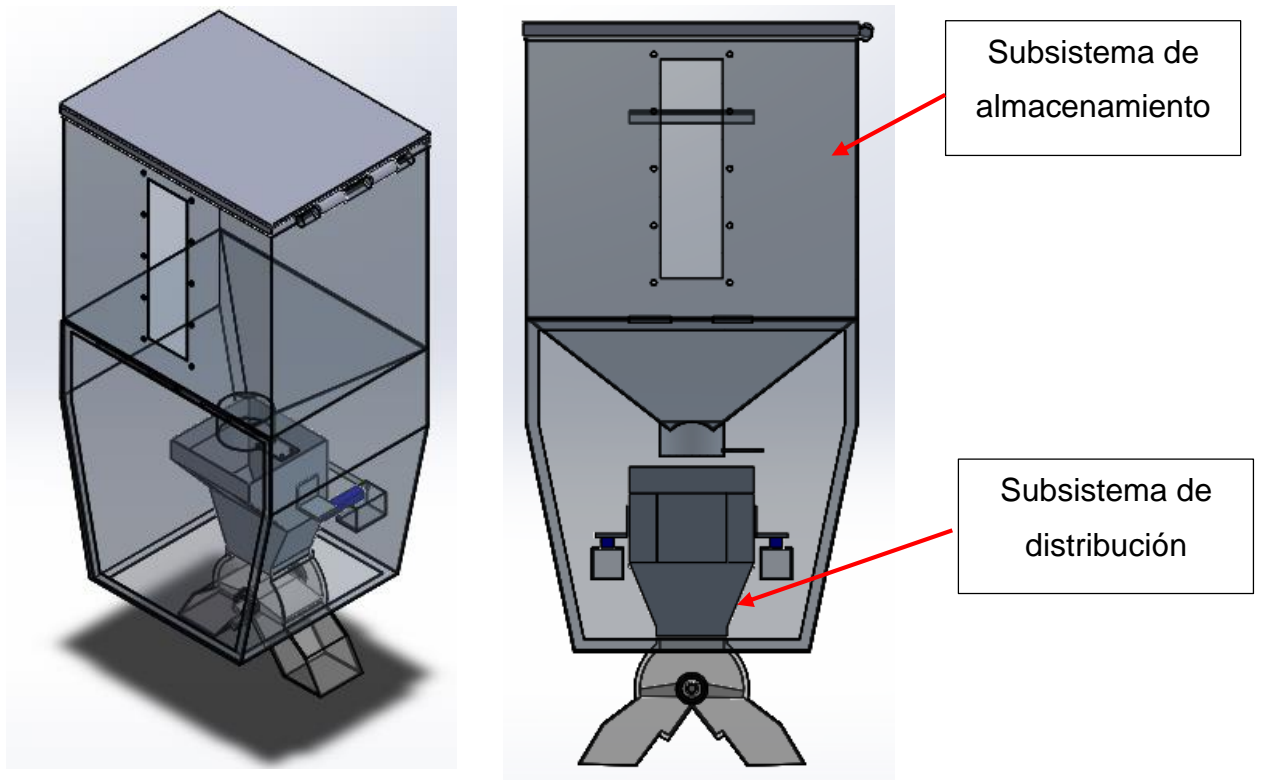
Fotografía 2. Prototipo de la alternativa 1.



Fuente: Autores

### 3.1.2.2. Alternativa 2

Figura 8. Alternativa 2 diseño del equipo.



Fuente: Autores

En este diseño al igual que en el anterior el almacenamiento es una sección prismática rectangular, se contempla el uso de dos celdas de carga para dar mayor estabilidad al recipiente de pesaje, en este caso la manera de distribuir el alimento es mediante una placa en la parte inferior encargada de entregar el alimento al equino correcto, como se observa en la figura 8 el modo de dirigir el alimento a los equinos es a través de tubería fabricada en lamina de acero inoxidable.

Este diseño no fue seleccionado debido a que el sistema de pesaje no era confiable en los datos entregados, respecto a la manufacturabilidad el diseño del subsistema de distribución requiere de gran cantidad de material en acero inoxidable y operaciones de fabricación agregado a lo anterior el cuerpo del dispensador tiene

una forma inclinada al final lo que dificulta la fabricación del equipo, lo anterior implica un incremento en los costos de fabricación y de material. (para ver más imágenes del prototipo ver anexo F)

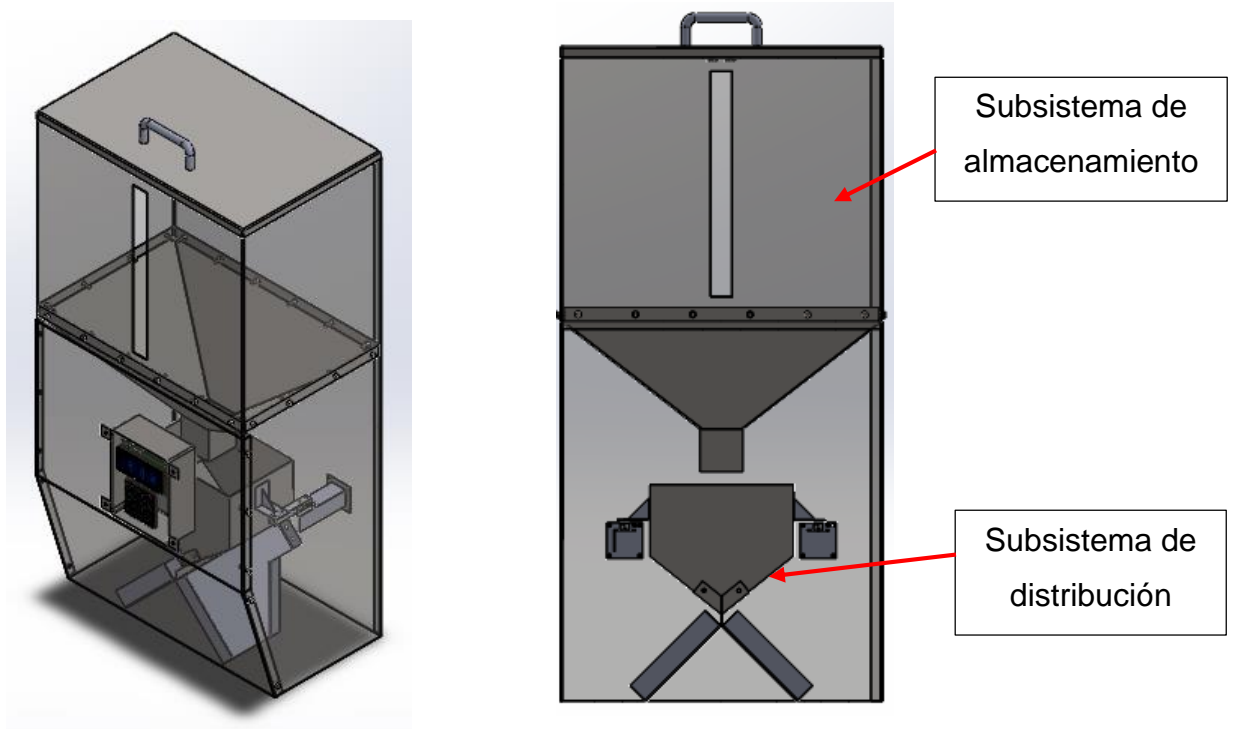
Fotografía 3. Prototipo de la alternativa 2.



Fuente: Autores

### 3.1.2.3. Alternativa 3

Figura 9. Alternativa 3 diseño del equipo



Fuente: Autores

La presente alternativa como las anteriores comparten el diseño de un subsistema de almacenamiento como una sección prismática rectangular. El subsistema de distribución consiste en dos celdas de carga para una mayor estabilidad y confiabilidad en los datos entregados al caer el alimento en el recipiente de pesaje, para entregar el alimento a los equinos este recipiente tiene dos compuertas que abre una o la otra mediante dos servomotores de acuerdo al equino a alimentar. La forma de dirigir el pienso es a través de una lamina doblada en inclinada para guiar el alimento al plato adecuado.

Se selecciono esta alternativa debido a las facilidades de manufacturabilidad y confiabilidad en la cantidad de alimento entregado al equino.

Una vez se eligió la alternativa de diseño y en base a los cálculos preliminares en los numerales 3.1.1 y 3.1.2, se estableció que las dimensiones de la tolva son:

*Sección prismática rectangular*

- Dimensión sección rectangular: 30 x 45 cm
- Altura 35 cm

*Sección de convergencia*

- Área mayor de 1350 cm<sup>2</sup>
- Área menor de 36 cm<sup>2</sup>
- Altura 15 cm

Con estas medidas iniciales se procedió a diseñar en detalle el dispensador de alimento (ver anexo B) y se elaboró el prototipo final del equipo (ver anexo G).

Fotografía 4. Prototipo final alternativa seleccionada.



Fuente: Autores

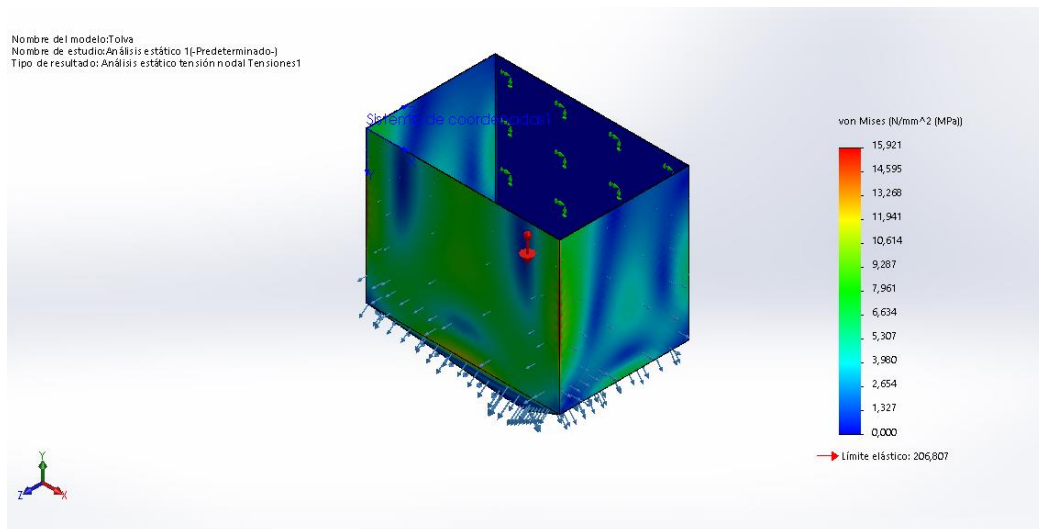
### 3.2. ANÁLISIS POR ELEMENTOS FINITOS

Para simular el comportamiento del equipo se utilizó el software SolidWorks, este mediante el complemento SolidWorks Simulation, permite hacer un análisis estructural de la pieza. Se generó un análisis estático lineal, los resultados obtenidos fueron las tensiones, deformaciones, desplazamientos y el factor de seguridad de la tolva. Los parámetros definidos para ejecutar el análisis fueron:

- Material: acero inoxidable 304
- Espesor: calibre 18
- Cantidad de alimento: 18 kg

**Tensiones:** El criterio de falla fue el esfuerzo de tensión Von Mises máximo debido a que el acero inoxidable es un material dúctil. El límite elástico del acero inoxidable 304 es de 206.807 MPa, el cual de acuerdo a la simulación no fue superado, es una tensión máxima de 15.921 MPa.

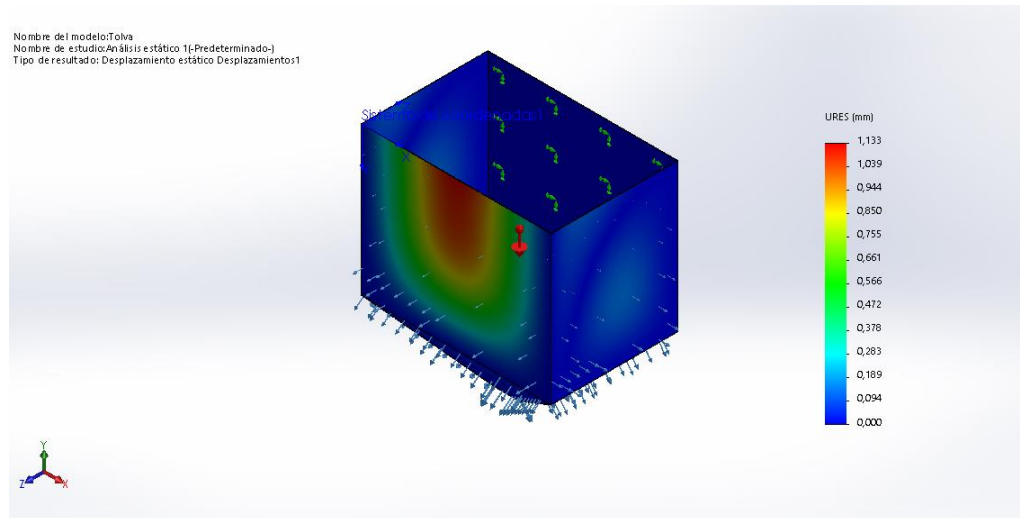
Figura 10. Tensiones análisis estático.



Fuente: Autores

**Desplazamientos:** para los desplazamientos el programa muestra donde va a haber y con cual valor. En este caso el valor máximo es de 1.133 mm y se encuentra en la zona más alejada de las sujeciones de la tolva.

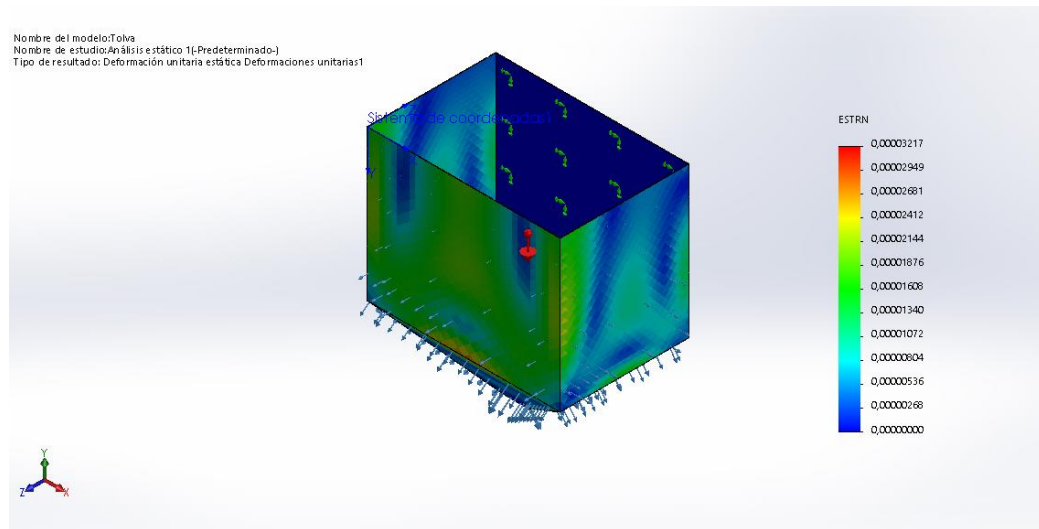
Figura 11. Desplazamiento análisis estático.



Fuente: Autores

**Deformaciones:** al realizar el análisis de deformaciones en el programa muestra el factor ESTRN el cual nos permite ver cómo se va a deformar el material y en qué zona. En este caso el máximo valor fue de 0,00001089.

Figura 12. Deformaciones análisis estático.



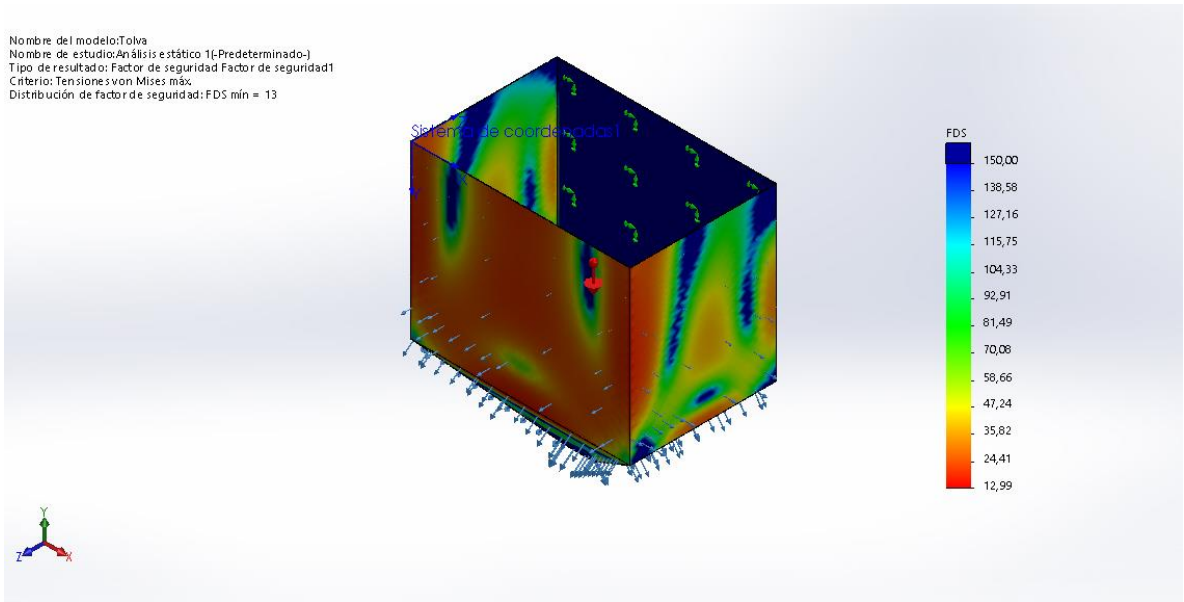
Fuente: Autores

**Factor de seguridad:** en un diseño mecánico se tiene en cuenta que funcione dentro de la zona elástica del material, esto para mantener su forma original, para estar seguros de cumplir con el ideal se usa un factor de seguridad el cual es la división entre el límite elástico del material y el esfuerzo máximo. Este valor debe ser superior a 1, de lo contrario indica que el diseño no soporta la carga bajo la cual está siendo sometido. De acuerdo a esto nuestro factor de seguridad debería ser:

$$\text{Factor de seguridad} = \frac{\sigma_y}{\sigma_{\text{VonMises}}} = \frac{206.807 \text{ MPa}}{15.921 \text{ MPa}} = 13$$

Con la herramienta de SolidWorks simulation se observó que efectivamente este factor de seguridad se cumple.

Figura 13. Factor de seguridad análisis estático.



Fuente: Autores

## 4. DISEÑO HARDWARE

Esta parte se encuentra separada por tres subsistemas como se indica en la figura 1. En la tabla a continuación se enuncian los componentes utilizados para el diseño de los subsistemas operacional y de control.

Tabla 4. Listado de elementos usados para el Hardware

Descripción	Cantidad	Subsistema
Arduino Mega	1	Control
Módulo conversor I2C para pantalla LCD	1	Control
Pantalla LCD 20x4	1	Control
Servomotor SG90	2	Operacional
Servomotor MG996R	1	Operacional
Teclado matricial 4x4 formato PCB 16 teclas	1	Control
Reloj DS3231	1	Control
Modulo Bluetooth HC05	1	Control
Celda de carga 1 Kg	2	Operacional
Sensor de nivel HC-SR04	1	Control
Modulo micro SD	1	Control
Módulo conversor I2C para teclado matricial	1	Control
Módulo Hx711	2	Operacional

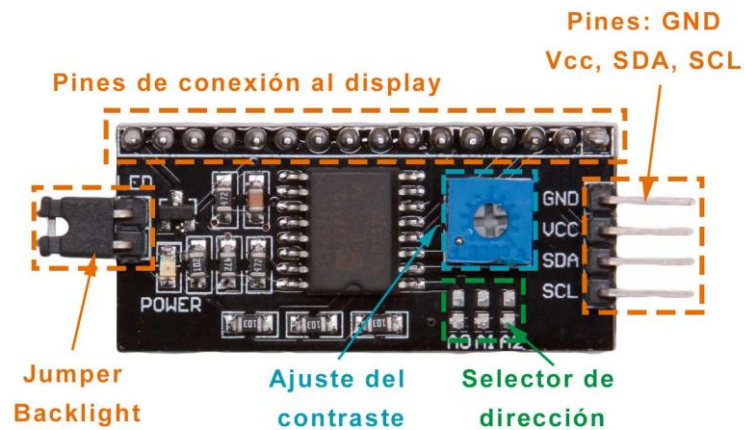
Fuente: Autores

### 4.1. SUBSISTEMA DE CONTROL

**4.1.1. Arduino MEGA 2560 R3.** Es un microcontrolador basado en ATmega2560, tiene 54 pines de entrada/salida digital, 15 pueden ser usados como salidas PWM, 16 son de entrada analógica, 4 UARTs (puertos seriales), un oscilador de cristal de 16MHz, puerto de electricidad, un header ICSP y un botón de reinicio. El rango de tensión de entrada se encuentra de 7V a 12V.



Figura 15. Pinout modulo conversor I2C.



Fuente: DIOSDADO, Raúl. Display LCD mediante bus I2C/TWI/SPI. Zona Maker, 2018. [En línea] [Consultado el 10 de enero de 2021] Disponible en: <https://www.zonamaker.com/arduino/modulos-sensores-y-shields/display-lcd-mediante-bus-i2c-twi-spi>

**4.1.3. Pantalla LCD 20x4.** Es una pantalla de cristal líquido que permite la visualización de caracteres en un espacio de 4 filas y 20 columnas, puede ser controlada por Arduino de dos maneras:

La primera es utilizando el bus de datos que se compone de 8 pines digitales, lo cual consume una gran cantidad de pines digitales del Arduino. La segunda opción es mediante un convertidor de estos pines a protocolo I2C esto reduce de manera significativa la cantidad de pines a solo dos (SDA 20 Y SCL 21). Cuenta con un pin para la calibración de brillo para ajustarlo por medio de un potenciómetro.

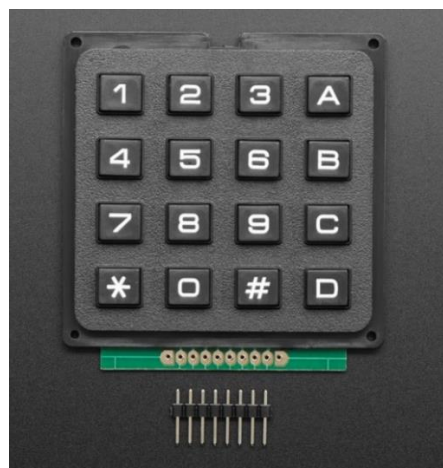
Figura 16. Pantalla LCD 20x4.



Fuente: BAKKER, Benne. How to control a character I2C LCD with Arduino. Makerguides.com. 2021. [En línea] [Consultado el 10 de enero de 2021] Disponible en: <https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/>

**4.1.4. Teclado matricial 4x4 formato PCB 16 teclas.** Las teclas se encuentran conectadas en matriz por lo que solo se requieren 8 pines del Arduino para leer las 16 teclas, sin embargo, la cantidad puede reducirse mediante un módulo conversor I2C de la misma forma que el usado para la pantalla LCD.

Figura 17. Teclado matricial 4x4.

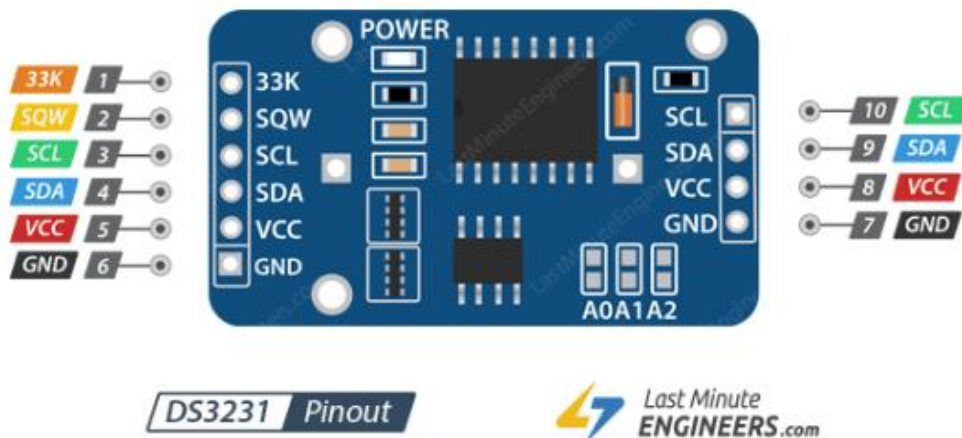


Fuente: ADAFRUIT. 4x4 Matrix Keypad. 2021. [En línea] [Consultado el 10 de enero de 2021] Disponible en: <https://www.adafruit.com/product/3844>

**4.1.5. Reloj DS3231.** Este módulo permite acceder a la hora y fecha incluso en condiciones de apagado, reiniciado o periodos largos de no alimentación del sistema. Cuenta con un chip Ds3231 muy preciso y de bajo costo. Con él se puede realizar aplicaciones de temporizadores, alarmas, relojes digitales cronometraje entre otras.

La fecha del mes se ajusta automáticamente para los meses con menos de 31 días, incluidos años bisiestos configurado hasta 2100. A nivel de hardware posee con un cristal de 32kHz para mantener ciclos de tiempo. Además, incluye una batería CR2032 y gracias a su bajo consumo de 3uA, suponiendo una batería de 220mAh facilitarían un funcionamiento mínimo de 8 años sin una fuente de alimentación externa.

Figura 18. Pinout reloj DS3231.



Fuente: LAST MINUTE ENGINEERS. Interface DS3231 Precision RTC Module with Arduino. 2021. [En línea] [Consultado el 11 de enero de 2021] Disponible en: <https://lastminuteengineers.com/ds3231-rtc-arduino-tutorial/>

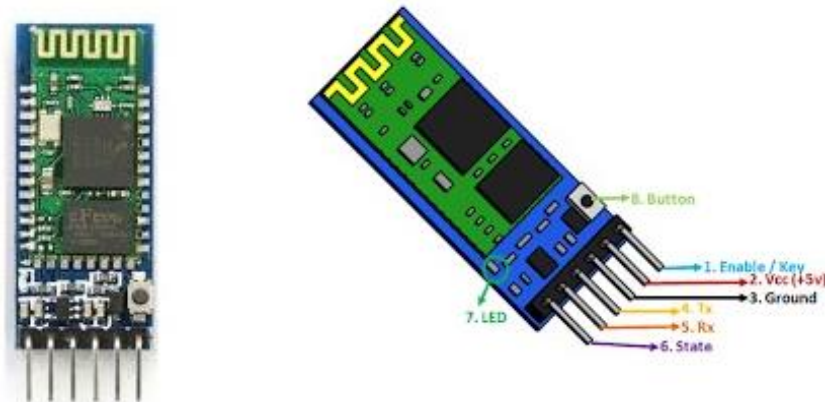
#### 4.1.6. Módulo Bluetooth HC-05

El bluetooth es una tecnología de comunicación inalámbricas, en la cual se usa la frecuencia de 2.4 GHz del espectro. Esta tecnología permite la transmisión de datos mediante radiofrecuencia, es muy usado en aplicaciones donde los dos dispositivos están relativamente cerca, hasta 10 metros dependiendo de la clase usada.

El módulo Bluetooth HC-05 es un dispositivo pensado para aplicaciones con la interfaz Arduino. Es de bajo consumo y posee la antena integrada. Puede ser conectado tanto como maestro o como esclavo y tiene un rango aproximadamente de 10 metros. Usa el protocolo de comunicación UART para conectarse con el microcontrolador Arduino.

Se alimenta con 5 voltios, pero los niveles seriales son de 3.3V. Se configura a partir de comandos AT enviados por conexión serial usando el mismo Arduino. Una vez configurado queda listo para que se usado en cualquier momento.

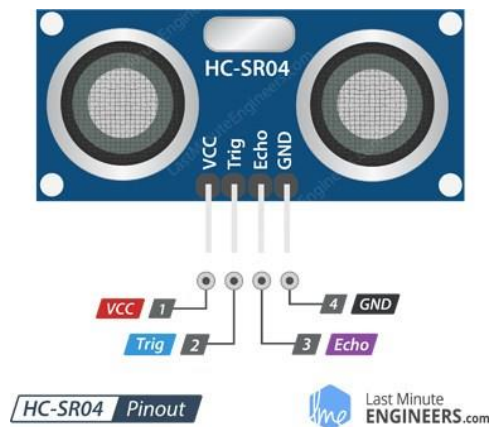
Figura 19. Pinout módulo HC05.



Fuente: ALI, Zahid. Introduction to HC-05. TheEngineeringProjects.com. 20 de octubre de 2019. [En línea] [Consultado el 11 de enero de 2021] Disponible en: <https://www.theengineeringprojects.com/2019/10/introduction-to-hc-05.html>

**4.1.7. Sensor ultrasonico HC-SR04.** Este sensor se encarga de enviar la señal de alarma cuando la cantidad de alimento es baja dentro del tanque de almacenamiento. Funciona midiendo la distancia con ondas ultrasonicas, emite un sonido y mide el tiempo que la señal demora en regresar. Trabaja en el aire y es posible que detecte objetos sin importar la forma, color, material y superficie. Tiene un rango minimo de 1.7 cm y un máximo de 450 cm, es de bajo consumo y alta precisión.

Figura 20. Pinout sensor ultrasónico HC-SR04.

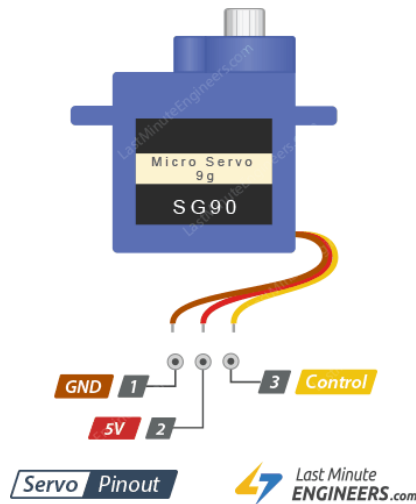


**Fuente:** LAST MINUTE INGEENIERS. How HC-SR04 Ultrasonic Sensor Works & Interface It with Arduino. 2021. [En línea] [Consultado el 11 de enero de 2021] Disponible en: <https://lastminuteengineers.com/arduino-sr04-ultrasonic-sensor-tutorial/>

## 4.2. SUBSISTEMA OPERACIONAL

**4.2.1. Servomotor SG90.** Como beneficios este servomotor es de bajo peso, tiene un bajo requerimiento de energía y tiene un excelente desempeño. El tipo de piñonería es plástica, con un torque de 1.8 kg-cm a 4.8V y un rango rotacional de 180°. Este servomotor va ubicado el subsistema de distribución, por lo que no se requiere un alto torque, se usaron dos de ellos.

Figura 14. Pinout servomotor SG90.



**Fuente:** LAST MINUTE ENGINEERS. How Servo Motor Works & Interface It With Arduino. 2021. [En línea] [Consultado el 12 de enero de 2021] Disponible en: <https://lastminuteengineers.com/servo-motor-arduino-tutorial/>

**4.2.2. Servomotor MG996R.** Este servomotor tiene una piñoria metálica, es ideal para ser ubicado donde se requiera alto torque. Puede rotar 120°, el torque es de 9.4 kg-cm a 4.8V y de 11 kg-cm a 6V. Se permite usar cualquier código, hardware o librerías para controlar este tipo de servomotor. Dentro del equipo va ubicado a la salida de la tolva de almacenamiento, es ideal por lo que se requiere contar con un alto torque para evitar la caída de alimento.

Figura 21. Conexiones servomotor MG996R



**Fuente:** COMPONENTES 101. MG996R Servo Motor. 2020. [En línea] [Consultado el 12 de enero de 2021] Disponible en: <https://components101.com/motors/mg996r-servo-motor-datasheet>

### 4.3. SUBSISTEMA ENERGÉTICO

**4.3.1. Adaptador Corriente Alterna (AC) / Corriente Directa (DC).** Teniendo en cuenta que la alimentación del equipo va a ser la red eléctrica, se requiere de un dispositivo que tome la corriente AC y la convierta a una corriente DC. El que se selecciono para el equipo tiene las siguientes características: entrada AC 100-240V 50/60 HZ 0.6 a salida DC 12V-2A.

Figura 17. Adaptador de AC/DC

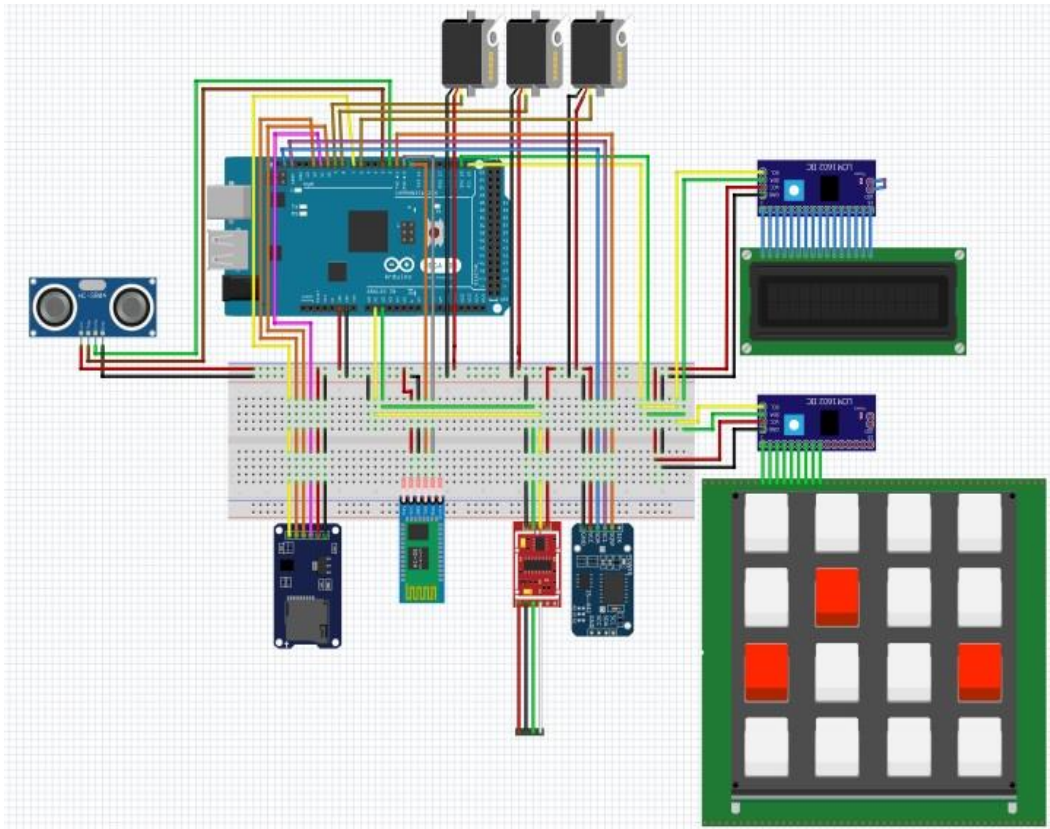


**Fuente:** ALIBABA.COM. Adaptador de alimentación de entrada de 100 de 240v AC 50/60hz 0.6a a DC 6V 6,5 V 100ma 300ma 350ma 450ma 500ma 800ma 1200ma 1500ma AC DC adaptador de potencia. 2021. [En línea] [Consultado el 13 de enero de 2021] Disponible en: <https://spanish.alibaba.com/product-detail/power-adapter-input-100-240v-ac-50-60hz-0-6a-to-dc-6v-6-5v-100ma-300ma-350ma-450ma-500ma-800ma-1200ma-1500ma-ac-dc-power-adapter-60839475759.html>

### 4.4. DIAGRAMA ESQUEMÁTICO MODULO DE CONTROL

A continuación, se muestra el diagrama esquemático del circuito correspondiente al sistema.

Figura 22. Diagrama esquemático



Fuente: Autores

## **5. DISEÑO SOFTWARE**

Con el avance de la tecnología y el crecimiento de las industrias se ha notado que existen tareas que hace una persona, las cuales, son repetitivas y se tiende a cometer errores ya sea por cansancio, exceso de confianza, entre otros.

Los sistemas de control es un conjunto de componentes que regulan el comportamiento de un sistema para lograr una acción específica, tienen la ventaja de poder hacer todas esas tareas sin cometer errores, con una gran precisión, se puede programar y saber con exactitud la cantidad de tareas que hará en un tiempo determinado, hacer varias tareas a la vez y demás.

### **5.1. SOFTWARE DE PROGRAMACIÓN**

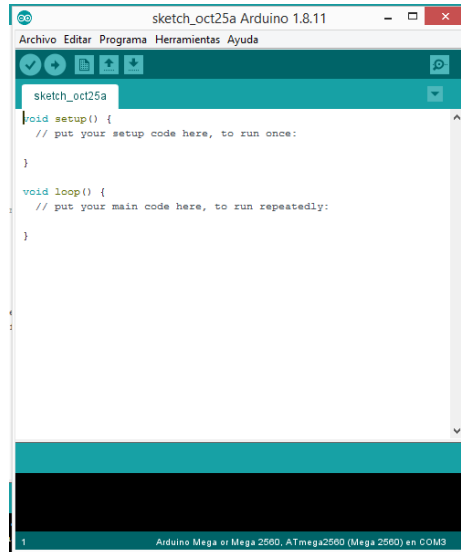
1, nació como una herramienta para realizar prototipos y equipos de poca potencia, para estudiantes que no tengan mucho conocimiento en electrónica y programación.

Debido al gran crecimiento que ha tenido se ha ido adaptando a proyectos más complejos en los cuales se pueden notar impresoras 3D, equipos de laboratorio, entre otras. Todas las placas de Arduino son de código abierto, en comparación con otros modos de control como lo es el PLC; sus componentes son bastante económicos al igual que los sensores que se le pueden adaptar.

#### **5.1.1. Descripción general**

Cada vez que se inicia un nuevo proyecto aparece la pantalla de la figura 23.

Figura 23. Pantalla inicial.



Fuente: Autores

Está dividida en dos partes void setup y void loop. El void setup solo se ejecutará una vez mientras que el void loop se ejecutará indefinidamente.

Figura 24. Void setup y void loop

```
Servo servoMotor1;
Servo servoMotor2;
Servo servoMotor3;
byte pinData = A1;//entrada de datos del sensor de peso
byte pinClk = A0;//señal de reloj del sensor de peso
const byte anzZeilen = 4; //filas del teclado
const byte anzSpalten = 4; //columnas del teclado
byte ale=0;
byte pant=0;byte caba=0;
bool edita=0;byte panta=0;
byte dig=0;byte pes=0; int ejecuta=0;int x=0;int y=0;int frec=0;int frec2=0;
byte pulsaciona=0; int seg=0;int des=0; int dis=240;int dis2=240;
int pesoac1[3]={2,0,0};int xdia[2]={0,9};int xdib[2]={0,4};
int pesoac2[3]={3,0,0};int distan=0;byte blu=0;int cambio=0;int mod=0;int mod2=0;
int hora=0;int minuto=0;int dia=0;int mes=0;int year=0; int i=0;
float pesoa1=1;float pesoa2=1;int modo=0;int dial=1;int dia2=1;float dia1=1;float dia2=1;
int cm=0;int z=0;int pos=0; int calibrar=0;int nivel=0;
//la funcion donde va la matriz del teclado
char tastenLayout[anzZeilen][anzSpalten] = {
    {'1','2','3','A'},
    {'4','5','6','B'},
    {'7','8','9','C'},
    {'*','0','#','D'}
};
```

Fuente: Autores

Arduino posee carpetas o archivos creados por personas con el objetivo de facilitar el uso de diferentes sensores pues evita tener que crear archivos de cada uno de ellos, esto es una gran ventaja al momento de usar un componente debido a que disminuye la cantidad de programación que puede ser larga y algo compleja.

Existen carpetas para el uso de la pantalla LCD, teclado matricial, bluetooth, servomotores, módulos i2c, sensor ultrasónico, entre otros muchos.

En este caso se usaron las siguientes carpetas:

Figura 25. Carpetas usadas

```
#include <NewPing.h>
#include <Keypad_I2C.h>
#include <Keypad.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <GFRTC.h>
#include <SD.h>
#include "HX711.h"
#include <Servo.h>
#include <SoftwareSerial.h>
```

Fuente: Autores

**5.1.2. Reloj DS3231.** Para el control del reloj se utilizó la programación presentada a continuación.

Figura 26. Programación para el control del reloj

```
struct timelib_tm datetime;
if (GFRTC.read(datetime)) {

    hora = datetime.tm_hour;
    minuto=datetime.tm_min;
    dia=datetime.tm_mday;
    mes=datetime.tm_mon;
    seg=datetime.tm_sec;
    year=datetime.tm_year -30;
} else {
    // error reading the RTC
    Serial.println(F("no se puede leer RTC."));
}
```

Fuente: Autores

**5.1.3. Alarma.** Para la generación de alarmas se utilizó el sensor ultrasónico HC-SR04 para indicar cuando el nivel de la tolva es insuficiente y se requiere llenar el equipo, cuando el sensor no detecta concentrado a una distancia de 30 cm cambia las variables que luego permitirán ver la alerta en la pantalla.

Figura 27. Generación de alarmas

```
if(cm<=30 && pant==0 && ale<2){ale=1;lcd.clear();nivel = 1;}
if(cm>30 && pant==0 && ale<2){ale=0;lcd.clear();nivel = 0;}
```

Fuente: Autores

Cuando la tolva este a una capacidad mínima aparecerá una pantalla de aviso, la cual se podrá desactivar presionando la tecla D, esto permitirá que se pueda realizar modificaciones a su funcionamiento. Si al momento de finalizar las modificaciones no se ha suministrado alimento a la tolva la pantalla de alerta volverá a aparecer y estará permanente hasta el momento en que se llene la tolva, cabe aclarar que esto no afectara el correcto funcionamiento de la máquina, ni interferirá al momento que se necesite realizar alguna modificación

**5.1.4. Pantalla LCD y teclado matricial.** Cada botón del teclado matricial se especificó en la programación de la siguiente manera dentro del void loop.

Figura 28. Teclado matricial

```
if (tecla=='1'){pes=1;}
if (tecla=='2'){pes=2;}
if (tecla=='3'){pes=3;}
if (tecla=='4'){pes=4;}
if (tecla=='5'){pes=5;}
if (tecla=='6'){pes=6;}
if (tecla=='7'){pes=7;}
if (tecla=='8'){pes=8;}
if (tecla=='9'){pes=9;}
if (tecla=='0'){pes=0;}
```

Fuente: Autores

Cada vez que se presiona un botón se guarda el valor en un vector el cual luego se convierte en un entero de la siguiente manera:

Figura 29. Vectores

```
pesoa1=(pesoac1[0]*100)+(pesoac1[1]*10)+(pesoac1[2]);
pesoa2=(pesoac2[0]*100)+(pesoac2[1]*10)+(pesoac2[2]);
dia1=(xdia[0]*10)+xdia[1];
dia2=(xdib[0]*10)+xdib[1];
```

Fuente: Autores

Jugando con los cambios de los valores de las variables se crearon las pantallas:

Figura 30. Creación de pantallas

```
if(pant==0 && panta==0 && ale==0){  
  
  lcd.setCursor(0,0);  
  lcd.print("      ! ALERTA ! ");  
  
  lcd.setCursor(0,1);  
  lcd.print("Llenar tolva");  
  lcd.setCursor(0,2);  
  lcd.print("Desactivar -> D");}  
  lcd.setCursor(0,3);  
  lcd.print("                ");  
  if(tecla == 'D' && ale==0){ale=2;lcd.clear();pant=0;panta=0; }
```

Fuente: Autores

**5.1.5. Comunicación Bluetooth.** Para lograr la comunicación entre Arduino y la aplicación Android, se usaron 5 variables.

Figura 31. Variables de comunicación

```
BT1.print('#'); //  
BT1.print(pesoal);  
BT1.print('A');  
BT1.print(dia11);  
BT1.print('B');  
BT1.print(peso2);  
BT1.print('C');  
BT1.print(dia22);  
BT1.print('D');  
BT1.print(nivel);  
BT1.print('~');
```

Fuente: Autores

Como hay variables que ocupan más de 2 bytes y Arduino no puede enviar valores enteros (int), se envían valores flotantes (float), del mismo modo desde la aplicación

se envían valores flotantes que al recibirlos, se deben convertir en vectores para que Arduino los pueda mostrar desde la pantalla LCD.

Figura 32. Conversión de vectores para pantalla LCD.

```
pesoac1[0]=pesoa1/100;
pesoac1[1]=pesoa1/10-pesoac1[0]*10;
pesoac1[2]=pesoa1-(pesoac1[0]*100+pesoac1[1]*10);
pesoac2[0]=pesoa2/100;
pesoac2[1]=pesoa2/10-pesoac2[0]*10;
pesoac2[2]=pesoa2-(pesoac2[0]*100+pesoac2[1]*10);

xdia[0]=dia1/10;
  if (xdia[0]>=1){
xdia[1]=dia1/1-xdia[0]*10;}
else {xdia[1]=dia1;}

xdib[0]=dia2/10;
  if (xdib[0]>=1){
xdib[1]=dia2/1-xdib[0]*10;}
else {xdib[1]=dia2;}
```

Fuente: Autores

**5.1.6. Programación raciones.** El equipo fue programado para funcionar de manera continua, entrega al equino la cantidad de porciones definidas por el usuario durante las veces programadas por el mismo. Se estableció un horario de funcionamiento entre las 4 am y las 10 pm.

Para que las porciones se entreguen en divisiones de tiempo iguales se usaron ecuaciones que son iguales para cualquier valor, aunque para los valores pares se agrega un proceso de más. Existen tres modos de funcionamiento:

1. Modo 1 = para el caballo 1
2. Modo 2 = para el caballo 2
3. Modo 3 = para ambos caballos

La programación es igual para ambos caballos solo cambia el nombre de las variables (dis1-dis2, dia1-dia2, frec1-frec2, ...), la programación del modo 3 es la combinación del modo 1 y modo 2.

Durante la entrega de alimento el servomotor gira de 0° a 90° aumentando de manera gradual, lo cual permite controlar el flujo de alimento por la salida de la tolva, al mismo tiempo las celdas de carga están calculando de manera intermitente para mayor precisión.

Hay dos fases durante el proceso, el primero es para que se va llenando el recipiente hasta el peso establecido menos 50 g, esto debido a que en ese momento está cayendo bastante alimento y de mantenerse así va a exceder el valor especificado. La segunda fase no va hasta el valor específico debido a que mientras cierra la salida la tolva, hay alimento que se encuentra cayendo y aún no ha sido pesado por las celdas de carga.

Cuando cumple la segunda fase, la salida de la tolva se cierra, dura 3 segundos y se abre la salida para el caballo, se mantiene 10 segundos, luego se cierra y sale del ciclo.

Figura 33. Programación de raciones

```
for (pos = 0; pos <= 90; pos += 1) {
servoMotor3.write(pos);
delay(800);
int pbasc=bascula.get_units();
Serial.print(pbasc,1);
Serial.print(" gramos");
Serial.println();

if(pbasc>=(pesoal-50) && pbasc<=2000 && i==1){pos=0; servoMotor3.write(pos); i=0;}
if(pbasc>=(pesoal-15) && pbasc<=2000){servoMotor3.write(0);
delay(3000);
servoMotor1.write(90);
delay(10000);break;}
  calibrar= pbasc;
}

dis=dis+frec;
}else{servoMotor1.write(0);servoMotor3.write(0);}

  if(modo==1){

mod=dial%2;

if (dis>1320){x=0;dis=240;}
if(cambio<240 || cambio>1320){des=1;dis=240;}else{

  if (dis<cambio){
    if(mod==0){
      frec=1080/dial;
      if(x==0){
        dis=dis+frec/2;x=1;
      }
      if(x==1){dis=dis+frec;}
    }else{

      frec=1080/(dial-1);
      dis=dis+frec;
    }
  }else{
    des=1;}
  }
}

//_____
if (mod==0 && des==1){
  frec=1080/dial;
  i=1;
```

```

if(x==0 && cambio==240){dis=dis+(frec/2);}

if(x==0 && cambio==dis){
  x=1;
  for (pos = 0; pos <= 90; pos += 1) {
    servoMotor3.write(pos);
    delay(800);
    int pbasc=bascula.get_units();
    Serial.print(pbasc,1);
    Serial.print(" gramos");
    Serial.println();
    if(pbasc>=(pesoa1-50) && pbasc<=2000 && i==1){pos=0; servoMotor3.write(pos); i=0;}
  if(pbasc>=(pesoa1-15) && pbasc<=2000){servoMotor3.write(0);
  delay(3000);
  servoMotor1.write(90);
  delay(10000);break;
  }
  }

  }else{servoMotor1.write(0);servoMotor3.write(0);}
if(x==1 && cambio==dis){dis=dis+frec;
  for (pos = 0; pos <= 90; pos += 1) {
    servoMotor3.write(pos);
    delay(800);

    int pbasc=bascula.get_units();
    Serial.print(pbasc,1);
    Serial.print(" gramos");
    Serial.println();
    if(pbasc>=(pesoa1-15) && pbasc<=2000 && i==1){pos=0; servoMotor3.write(pos); i=0;}
  if(pbasc>=(pesoa1) && pbasc<=2000){servoMotor3.write(0);
  delay(3000);
  servoMotor1.write(90);
  delay(10000);break;
  }
  }

  }

  }else{servoMotor1.write(0);servoMotor3.write(0);}

```

```

if (mod!=0 && des==1){
  i=1;
  frec=1080/(dia1-1);
  if(cambio==dis){
    for (pos = 0; pos <= 90; pos += 1) {
      servoMotor3.write(pos);
      delay(800);
      int pbasc=bascula.get_units();
      Serial.print(pbasc,1);
      Serial.print(" gramos");
      Serial.println();

      if(pbasc>=(pesoa1-50) && pbasc<=2000 && i==1){pos=0; servoMotor3.write(pos); i=0;}
      if(pbasc>=(pesoa1-15) && pbasc<=2000){servoMotor3.write(0);
      delay(3000);
      servoMotor1.write(90);
      delay(10000);break;}
    //   calibrar= pbasc;
    }

    dis=dis+frec;
  }else{servoMotor1.write(0);servoMotor3.write(0);}

}

```

Fuente: Autores

**5.1.7. Calibración celdas de carga.** Para realizar la medición del alimento se utilizó dos celdas de carga, las cuales traducen la presión o fuerza en una señal eléctrica. Las celdas utilizan un amplificador HX711 el cual se encarga de transmitir la información de la celda al microcontrolador, al inicio para su uso correcto deben ser calibradas.

Como primer paso se incluyó la librería "HX711.h" en el código, se seleccionan los pines para salida de datos y de reloj. Para calibrar primero se requiere tarar las celdas, una vez se hace ya se puede iniciar el proceso de calibración, para esto se requiere tener un peso conocido. Cuando se inicie el proceso se debe aumentar o

disminuir el factor de calibración hasta llegar al valor de masa buscado, esto nos dará el factor de calibración deseado.

Figura 34. Calibración de celdas de carga

```
CALIBRAR_CELDAS $
#include "HX711.h"

HX711 bascula;

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

int x=1;

// Parámetro para calibrar el peso y el sensor
float factor_calibracion = 295;

void setup() {
  Serial.begin(9600);
  Serial.println("HX711 programa de calibracion");
  Serial.println("Quita cualquier peso de la bascula");
  Serial.println("Una vez empiece a mostrar informacion de medidas, coloca un peso conocido encima de la bascula");
  Serial.println("Presiona + para incrementar el factor de calibracion");
  Serial.println("Presiona - para disminuir el factor de calibracion");

  // Iniciar sensor
  bascula.begin(pinData, pinClk);

  // Aplicar la calibración
  bascula.set_scale();
  // Iniciar la tara
  // No tiene que haber nada sobre el peso
  bascula.tare();
}

CALIBRAR_CELDAS $
// Obtener una lectura de referencia
long zero_factor = bascula.read_average();
// Mostrar la primera desviación
Serial.print("Zero factor: ");
Serial.println(zero_factor);
}

void loop() {

  if (x=1){
    // Aplicar calibración
    bascula.set_scale(factor_calibracion);

    // Mostrar la información para ajustar el factor de calibración
    Serial.print("Leyendo: ");
    Serial.print(bascula.get_units(), 1);
    Serial.print(" gs");
    Serial.print(" factor_calibracion: ");
    Serial.print(factor_calibracion);
    Serial.println();
  }
  // Obtener información desde el monitor serie
  if (Serial.available())
  {
    char temp = Serial.read();
    if (temp == '+')
      factor_calibracion += 10;
    else if (temp == '-')
      factor_calibracion -= 10;
  }
}
```

Fuente: Autores

**5.1.8. Conservación de datos.** El dispensador tiene la capacidad de guardar la información en caso de alguna falla eléctrica o si a persona desea revisar el historial de alimentación de los equinos. Para esto se utilizó una memoria micro SD. En el código Arduino se debió agregar la librería mostrada a continuación para su uso.

```
#include <SD.h>
```

Posteriormente se procedió a definir el pin digital con el cual se conectó la memoria y se creó la carpeta en la cual se encontrará almacenada la información, esta se encuentra en un documento TXT.

Todo esto antes de entrar al void setup, para guardar información se utilizó el siguiente código, el cual va en la pantalla ejecutando.

Figura 35. Conservación de datos

---

```
if (pant==4 && ale>=1) {
  lcd.setCursor(0,1);
  lcd.print("Ejecutando ... ");
  delay(2000);
  pant=0;panta=0;caba=0;ale=0;
  modo=1;

  //*****
  archivo = SD.open("prueba.txt", FILE_WRITE);
  if (archivo) {

    archivo.print(dia);
    archivo.print("/");
    archivo.print(mes);
    archivo.print("/");
    archivo.print(year);
    archivo.print(" ");
    archivo.print(hora);
    archivo.print(":");
    archivo.print(minuto);
    archivo.print(" ");
    archivo.print("Caballo 1");
    archivo.print(" ");
    archivo.print("Cantidad ");
    archivo.print(pesoal);
    archivo.print(" / ");
    archivo.print("Veces por dia ");
    archivo.println(dial);
```

```

Serial.println("Escribiendo en archivo prueba.txt..."); // texto en monitor serie
archivo.close(); // cierre del archivo
Serial.println("escritura correcta"); // texto de escritura correcta en monitor serie
} else {
Serial.println("error en apertura de prueba.txt"); // texto de falla en apertura de archivo
}

archivo = SD.open("prueba.txt"); // apertura de archivo prueba.txt
if (archivo) {
Serial.println("Contenido de prueba.txt:"); // texto en monitor serie
while (archivo.available()) { // mientras exista contenido en el archivo
Serial.write(archivo.read()); // lectura de a un caracter por vez
}
archivo.close(); // cierre de archivo
} else {
Serial.println("error en la apertura de prueba.txt");// texto de falla en apertura de archivo
}
//*****
}

```

Fuente: Autores

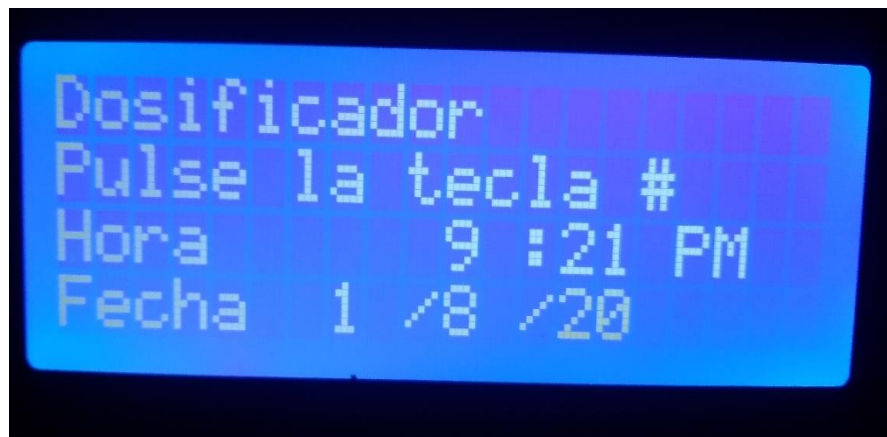
Aparece en el siguiente orden (Día/mes/año Hora: minuto caballo 1 cantidad de alimento/veces por día), en caso que se programe para los dos caballos, seguido de caballo 1 aparecerá las modificaciones de caballo 2 en el mismo renglón.

## 5.2. INTERFAZ HUMANO – MÁQUINA

El dispensador se encuentra diseñado para ser controlado de manera local y remota.

**5.2.1. IHM Local.** Cuenta con una pantalla principal en la cual se podrá ver la hora y fecha, se debe tener en cuenta que para avanzar en algunos casos se debe presionar “#” y para regresar en cualquier momento a la pantalla principal se debe presionar " \* "

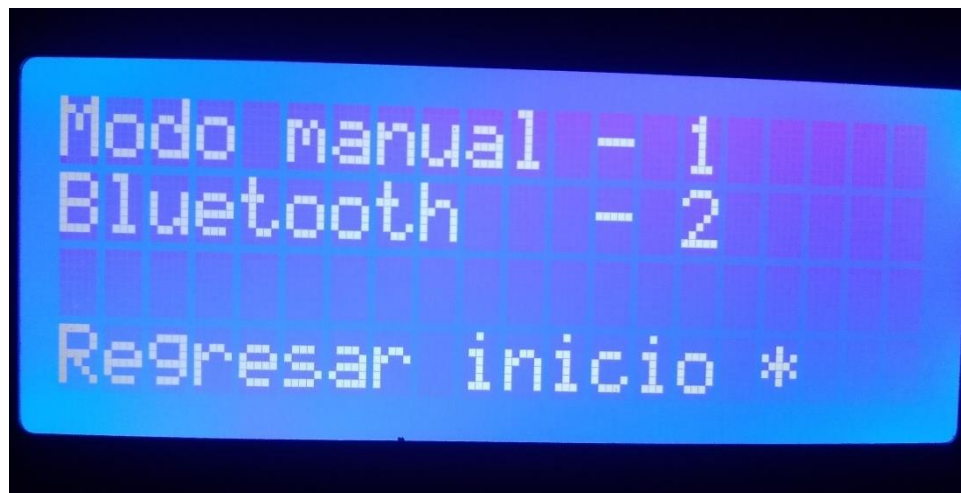
Figura 36. IHM Local. Hora y fecha



Fuente: Autores

Al presionar “#” pasa a la siguiente pantalla, en este punto se puede seleccionar de qué manera se modificaran los valores, en modo manual o por medio de la aplicación Android, en este caso modo manual.

Figura 37. IHM Local. Modo manual

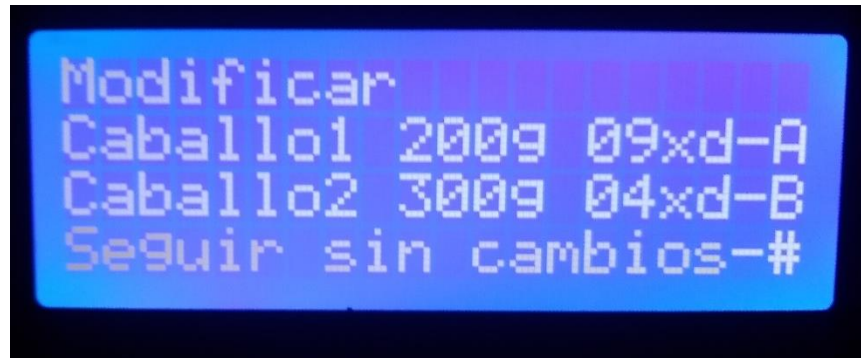


Fuente: Autores

En esta pantalla se puede divisar los valores de cantidad de alimento y veces por día de ambos caballos, a su vez de ser necesario modificar dichos valores

solamente se debe presionar la tecla específica para esa función (A, B), si no es el caso y se piensa dejar los valores presentes sin cambios se presiona “#”.

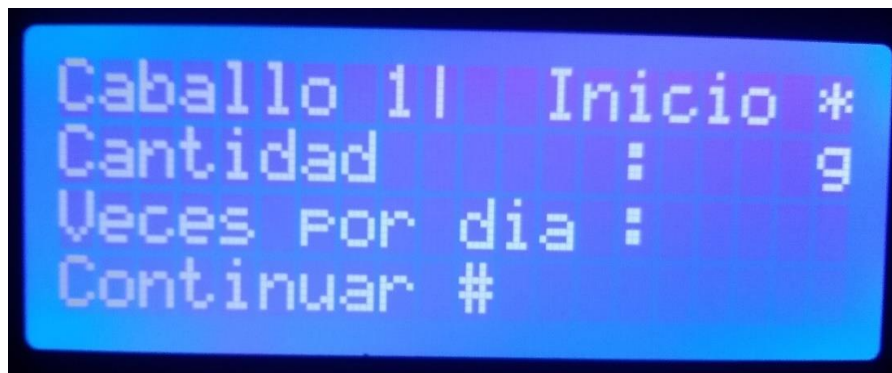
Figura 38. IHM Local. Modificación de valores



Fuente: Autores

En este caso se seleccionó la opción de modificar valores para el caballo 1, después presionar # para seguir a la nueva pantalla.

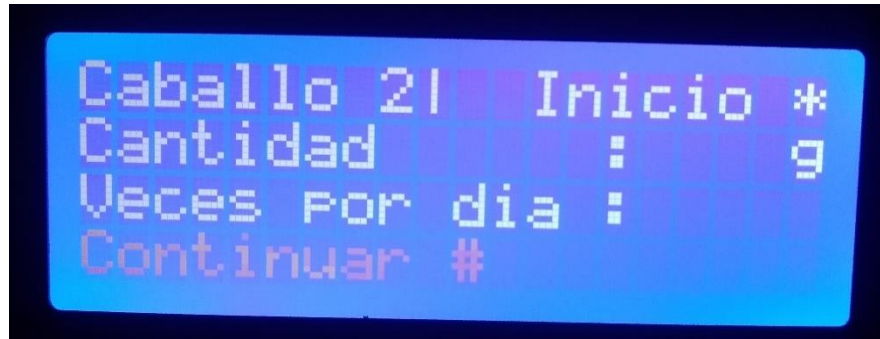
Figura 39. IHM Local. Modificación de valores, caballo 1.



Fuente: Autores

Si se seleccionó el caballo 2 aparecerá la siguiente pantalla

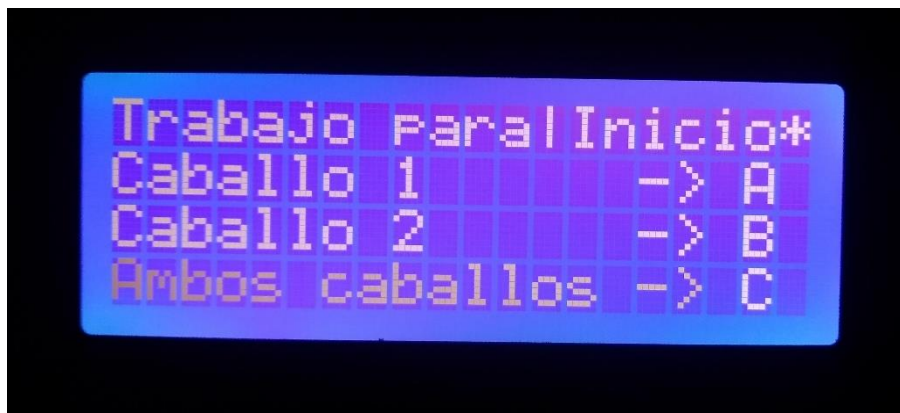
Figura 40. IHM Local. Modificación de valores, caballo 2.



Fuente: Autores

Acá se puede elegir cualquiera de las funciones de preferencia, en el momento que se encuentre un solo caballo se selecciona la opción caballo 1 o caballo 2 dependiendo de la salida del alimento en la máquina, si es el caso que los dos estén presentes se selecciona la opción de ambos caballos; hay que tener en cuenta que para varios valores de veces por día se debería suministrar el alimento para los dos caballos a la misma hora, lo cual es imposible por el sistema de la máquina que debe entregar el alimento uno a la vez, así que en caso que pasara esto, se entregará el alimento al caballo 1 y luego de acabada esta función se entregará al caballo 2.

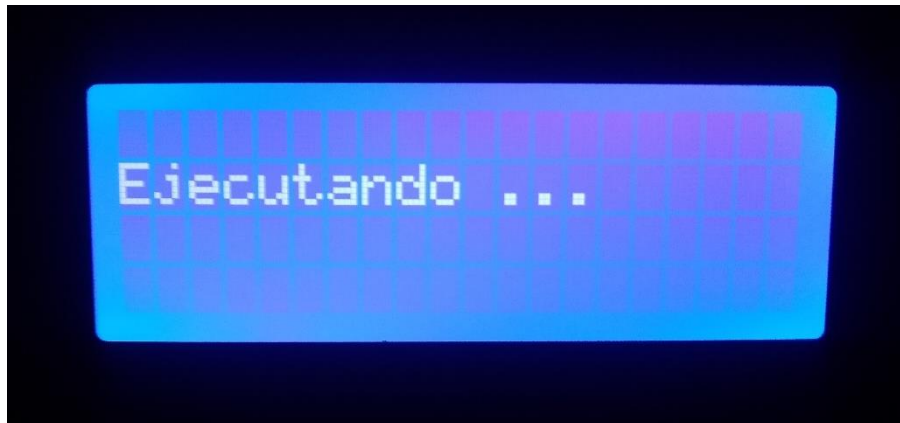
Figura 41. IHM Local. Opciones de preferencia.



Fuente: Autores

A continuación de haber seleccionado cualquiera de las opciones anteriores aparecerá la pantalla de ejecutando, la cual confirmará que la modificación quedó guardada y se ejecutará.

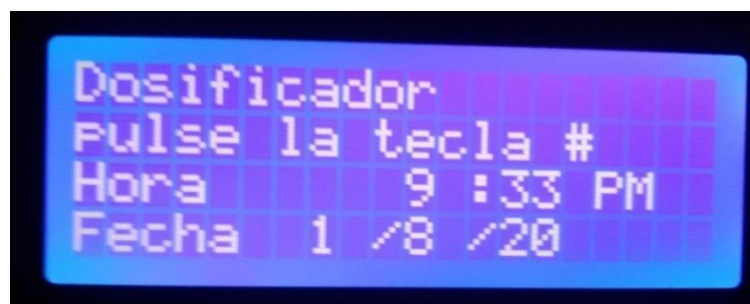
Figura 42. IHM Local. Ejecutando.



Fuente: Autores

La pantalla ejecutando tarda 2 segundos, luego de esto regresará a la pantalla principal.

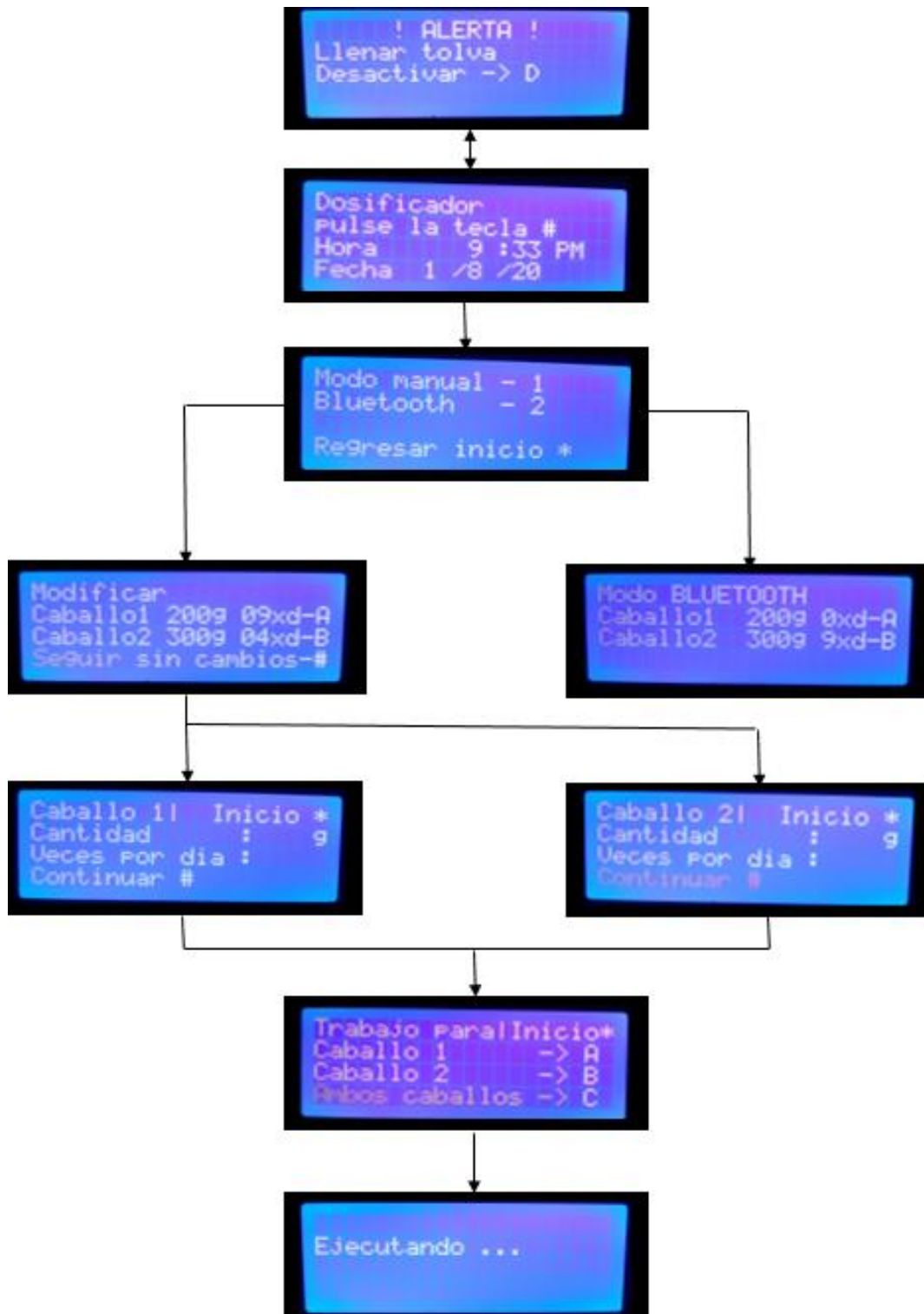
Figura 43. IHM Local. Pantalla principal



Fuente: Autores

Como resumen del funcionamiento de la interfaz podemos observar el diagrama de flujo, a continuación.

Figura 44. Diagrama de flujo pantallas en LCD.

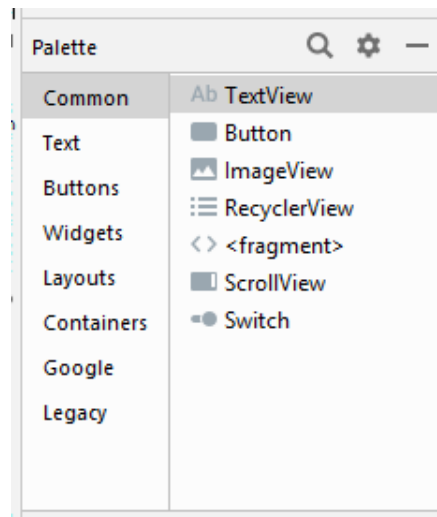


Fuente: Autores

**5.2.2. IHM Remota.** Para la interfaz remota se creó una aplicación mediante Android Studio, el cual es el programa oficial de Android para el desarrollo de aplicaciones y proporciona muchas facilidades a la hora de desarrollar una App.

Dentro del programa se puede elaborar la interfaz por medio de un código o a través de la vista previa del design. En este caso por simplicidad y fácil manejo se llevó a cabo mediante el Design y con el uso del Pallette que es una barra de herramientas para el manejo del diseño gráfico.

Figura 45. IHM Remota. Interfaz



Fuente: Autores

Cada vez que se agrega una herramienta de manera automática se crea un archivo para cada Layout en donde se encontrará la programación de la sección gráfica.

Cuando se crea un Layout automáticamente se crea un archivo java con el mismo nombre, en el cual está la programación de las acciones a realizar por la app en esa pantalla.

Figura 46. IHM Remota.Programación.



```
1 package com.dosificadora;
2
3 import androidx.appcompat.app.AppCompatActivity;
4
5 import android.content.ContentValues;
6 import android.database.Cursor;
7 import android.database.sqlite.SQLiteDatabase;
8
9
10 import android.content.Intent;
11 import android.widget.EditText;
12 import android.os.Bundle;
13 import android.view.View;
14 import android.widget.Toast;
15
16
17 public class RegistroActivity extends AppCompatActivity {
18
19     EditText NameUser, User,Pasport;
20     private boolean Bandera1, Bandera2;
21
22     @Override
23     protected void onCreate(Bundle savedInstanceState) {
24         super.onCreate(savedInstanceState);
25         setContentView(R.layout.activity_registro);
26
27         NameUser= (EditText)findViewById(R.id.Nombre);
28         User= (EditText)findViewById(R.id.Usuario);|
```

Fuente: Autores

A continuación se menciona algunas de las funciones principales de Android Studio.

**Text View:** agrega texto que es visible pero no se puede modificar, se usó para agregar texto a las diferentes pantallas (Layout) como es el caso de usuario, contraseña, registro, nombre, contraseña anterior, contraseña nueva, dispositivos vinculados, caballo 1, caballo 2, cantidad de alimento, veces por día, Alerta. Para el caso de Alerta es un TextView que aparece como mensaje emergente cuando el

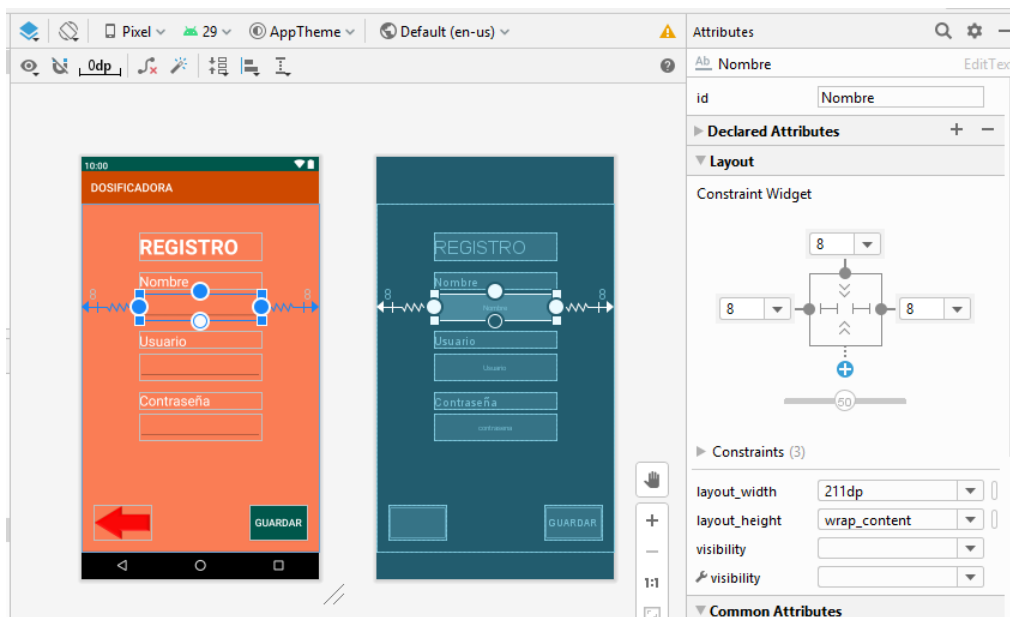
nivel de la tolva es bajo, además de este hay más mensajes emergentes como contraseña incorrecta, usuario incorrecto, debes llenar todos los campos.

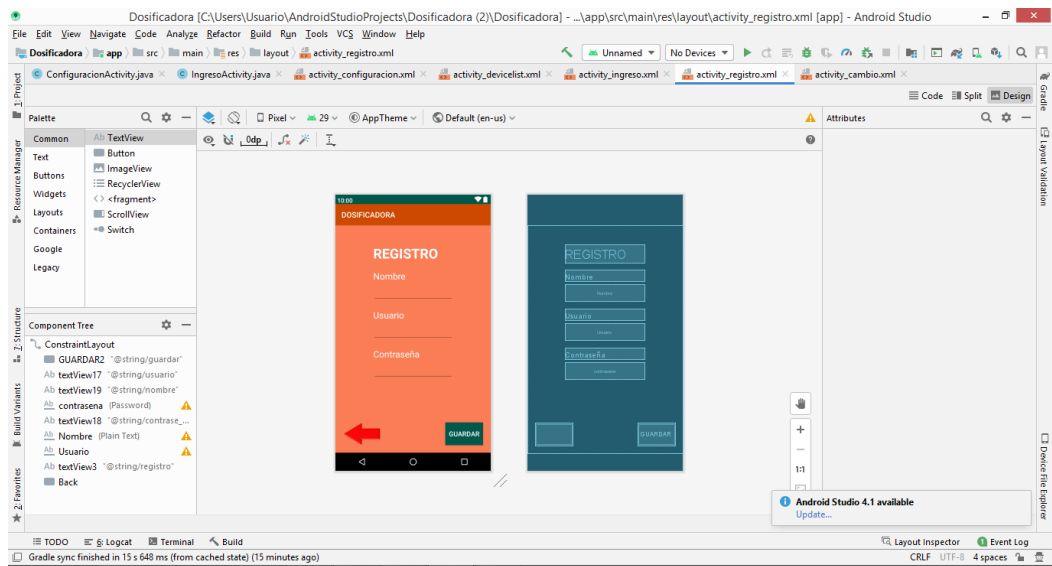
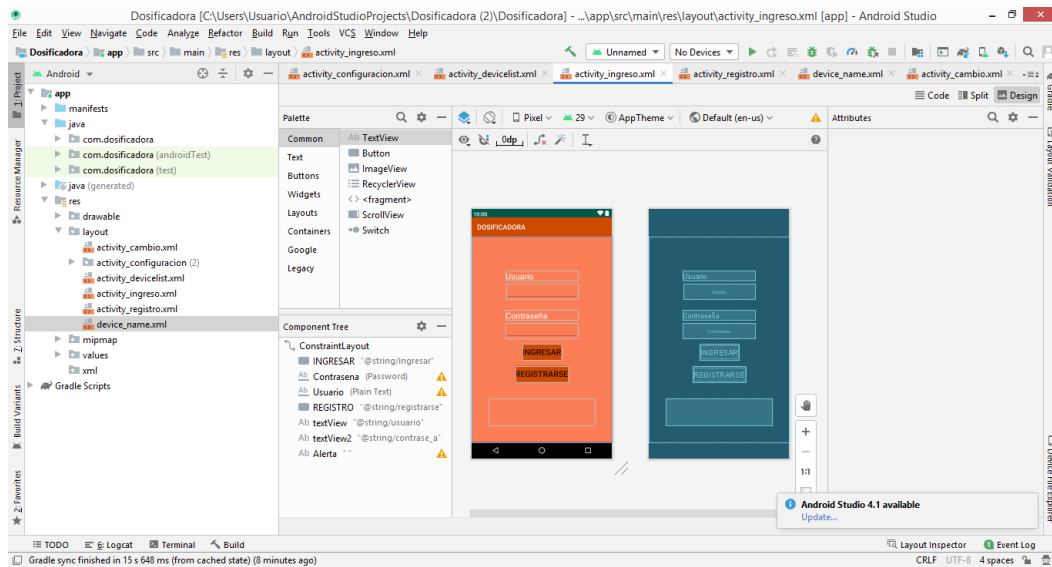
**EditText:** agrega un cuadro de texto en el cual se pueden poner valores numéricos y escribir texto a convenir, se usó para poder agregar los valores a modificar de cantidad de alimento, veces por día, usuario, contraseña, nombre.

**Button:** agrega botones, los cuales se usaron para poder realizar la función, esta muestra texto que en este caso no se modifica, pero realiza una acción al presionarla, se usó para los botones ingresar, registrarse, guardar, la flecha de regreso, leer, cambiar contraseña, calibrar.

Se debe tener en cuenta el id (identificación) de cada herramienta debido a que esta dirección es requerida para la comunicación entre pantallas.

Figura 47. IHM Remota. Funciones principales.





Fuente: Autores

Una vez se desarrolló la aplicación, la interfaz de ingreso se diseñó de la siguiente forma, en el lado izquierdo se ve desde el punto de vista del usuario y el izquierdo como desarrollador de la aplicación.

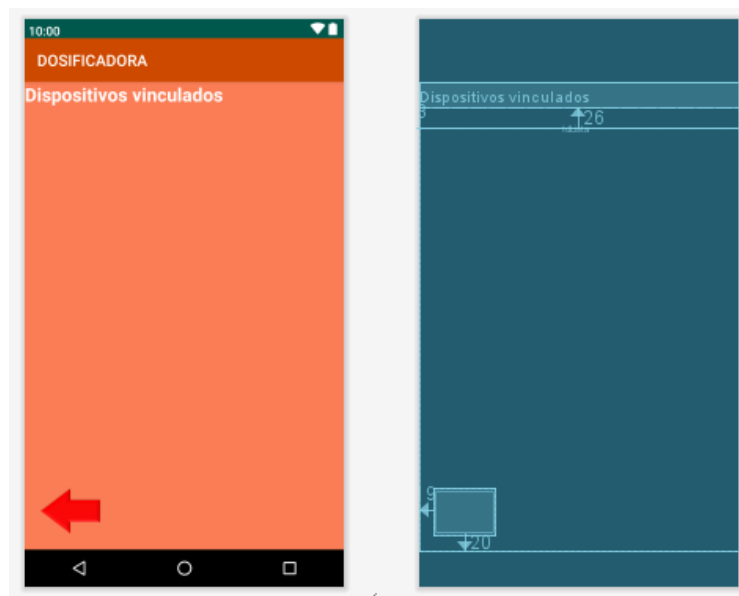
Figura 48. IHM Remota. Interfaz de ingreso.



Fuente: Autores

Para ingresar a la aplicación por primera vez el usuario debe registrarse y crear su cuenta. A continuación, deberá vincular su Smartphone con el equipo.

Figura 49. IHM Remota. Ingreso.



Fuente: Autores

En la aplicación el usuario podrá configurar la cantidad de alimento a entregar al equino y las veces por días.

Figura 50. IHM Remota. Configuración de la cantidad de alimento



Fuente: Autores

En caso de que se modifiquen los datos por medio de la aplicación, se pueden divisar al abrir la pantalla de modo bluetooth.

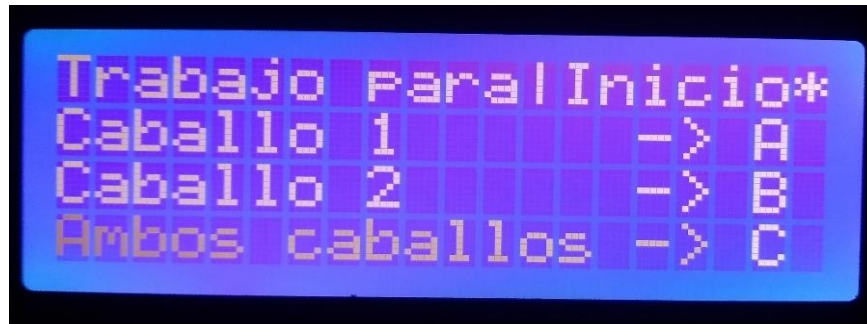
Figura 51. Cambio en la pantalla de modo bluetooth



Fuente: Autores

Al seleccionar # se podrá avanzar a la siguiente pantalla, en la cual da la posibilidad de seleccionar si se quiere que le suministre alimento a un caballo, a el otro caballo o a ambos.

Figura 52. Selección de suministro de alimento



Fuente: Autores

## 6. PRUEBAS Y RESULTADOS

El dispensador puede ser programado por el usuario de forma local y remota. Para ser programado localmente se usa un teclado matricial 4x4 y una pantalla LCD 16x4. De manera remota se usa una aplicación para el sistema Android donde el usuario puede programar la cantidad y horarios de entrega del alimento. En este capítulo se muestran los resultados de las pruebas realizadas al dispensador.

Se realizó de la siguiente forma: se hicieron dos pruebas para el equino 1 y dos pruebas para el equino 2, en las cuales se varió el peso, pero se dejó la misma cantidad de veces al día 5 para el equino 1 y 6 para el equino 2, en estas pruebas se verificó que dispensara la cantidad de veces especificada, a la hora indicada por la tabla de pruebas y la cantidad de alimento que se espera, a su vez se denotó el funcionamiento de la alarma que avisa cuando la tolva tiene de bajo nivel de alimento.

El video funcional del equipo y las fotografías puede verse en los anexos I al xx.

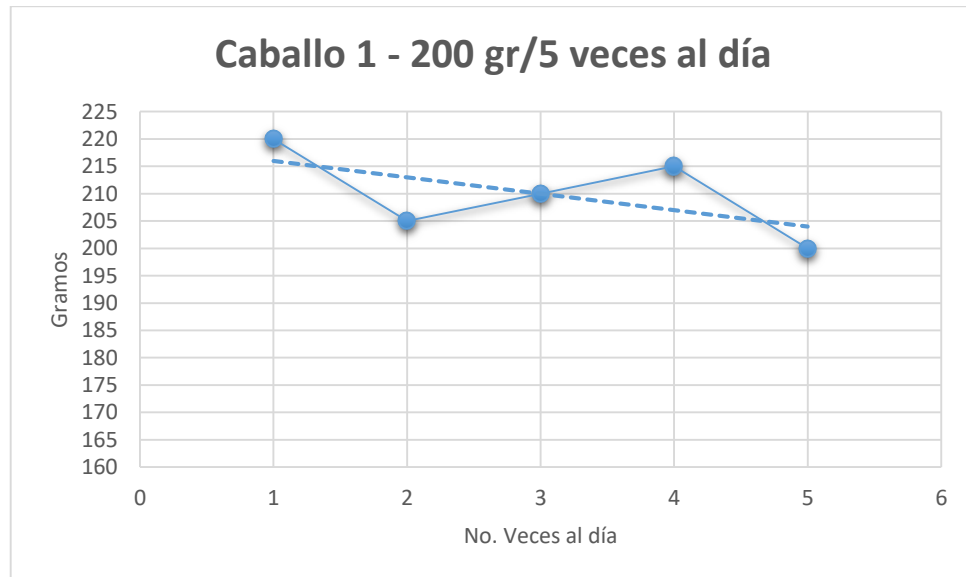
### 6.1. PROGRAMACIÓN LOCAL PARA DISPENSAR EL ALIMENTO A UN EQUINO

Tabla 5. Cantidad de alimento entregado al equino 1.

<b>Caballo 1 - 200 gr</b>	
Veces al día	Gramos
1	220
2	205
3	210
4	215
5	200
<b>Promedio</b>	210

Fuente: Autores

Gráfica 1. Cantidad de alimento entregado al equino 1.



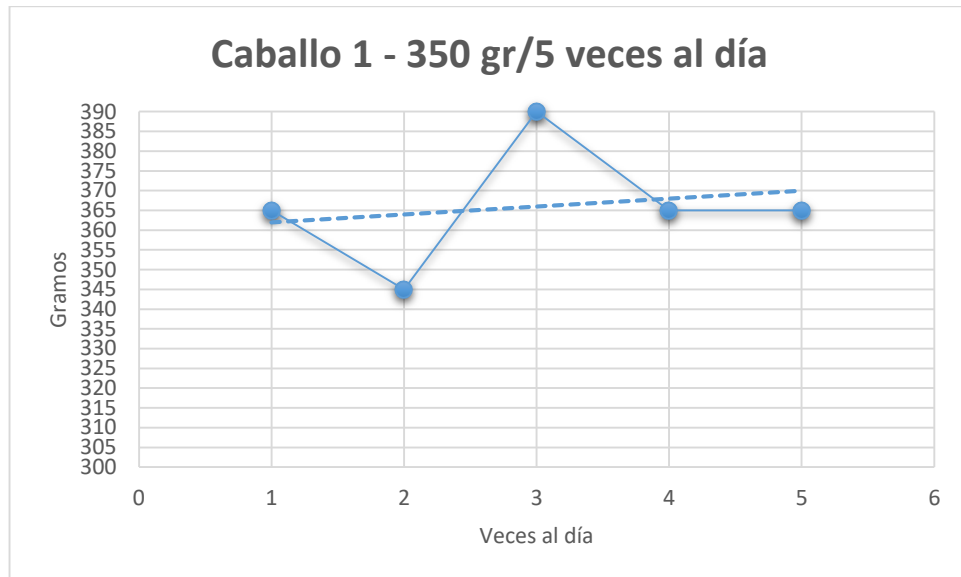
Fuente: Autores

Tabla 6. Cantidad de alimento entregado al equino 1.

<b>Caballo 1 - 350 gr</b>	
Veces al día	Gramos
1	365
2	345
3	390
4	365
5	365
<b>Promedio</b>	<b>366</b>

Fuente: Autores

Gráfica 2. Gráfica cantidad de alimento entregado al equino 1.



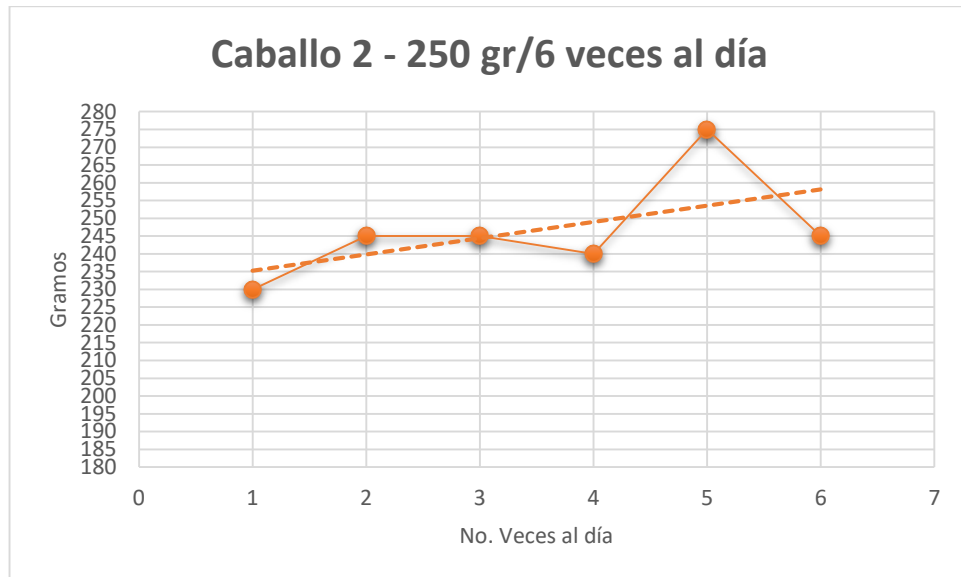
Fuente: Autores

Tabla 7. Alimento entregado al equino 2.

<b>Caballo 2 - 250 gr</b>	
Veces al día	Gramos
1	230
2	245
3	245
4	240
5	275
6	245
<b>Promedio</b>	<b>246.67</b>

Fuente: Autores

Gráfica 3. Cantidad de alimento entregado al equino 2.



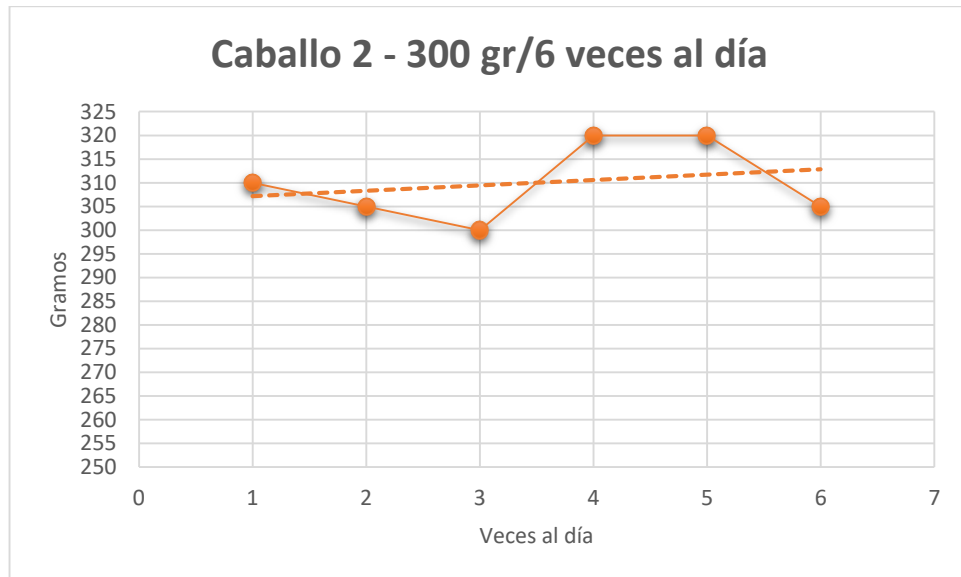
Fuente: Autores

Tabla 8. Cantidad de alimento entregado al equino 2.

<b>Caballo 2 - 300 gr</b>	
Veces al día	Gramos
1	310
2	305
3	300
4	320
5	320
6	305
<b>Promedio</b>	<b>310</b>

Fuente: Autores

Gráfica 4. Cantidad de alimento entregado al equino 2.



Fuente: Autores

## **7. MANUAL DEL USUARIO**

En este se dan a conocer las recomendaciones para el correcto uso de la máquina, cuidado y operación del dispensador automático para equinos:

- Advertencias
- Componentes
- Inicio del equipo (modo manual / modo remoto)
- Alertas del equipo

Se adjunta en un archivo en formato pdf en el anexo H.

## 8. COSTOS DEL EQUIPO

Para el cálculo de los costos del proyecto se tuvo en cuenta el costo de los materiales mecánicos, componentes electrónicos, y asesorías. En la tabla 4-7 se registra el presupuesto para la construcción del prototipo.

Tabla 9. Costos del equipo.

CANT	DESCRIPCIÓN	VALOR	TOTAL
1	Pantalla LCD 4x20	\$ 27.500	\$ 27.500
2	Servomotor SG90	\$ 13.000	\$ 26.000
1	Modulo Bluetooth HC05	\$ 32.000	\$ 32.000
1	Módulo I2C para pantalla LCD	\$ 8.235	\$ 8.235
1	RTC-DS1302 Reloj	\$ 7.563	\$ 7.563
1	Batería de cronometro	\$ 1.681	\$ 1.681
2	Celda de carga + Hx711	\$ 16.555	\$ 33.110
1	Sensor ultrasónico	\$ 7.000	\$ 7.000
1	Teclado matricial 4x4	\$ 15.000	\$ 15.000
1	Adaptador corriente AC/DC	\$ 12.000	\$ 12.000
1	Adaptador memoria SD	\$ 7.000	\$ 7.000
1	Memoria SD	\$ 16.000	\$ 16.000
1	Alambre para conexión macho, hembra	\$ 7.000	\$ 7.000
1	Servomotor MG996R	\$ 23.000	\$ 23.000
1	Arduino Mega 2560 R3 con Cable USB - Compatible	\$ 45.000	\$ 45.000
1	Costos prototipo alternativa 1	\$ 25.000	\$ 25.000
1	Costos prototipo alternativa 2	\$ 15.000	\$ 15.000
2	Costos prototipo alternativa 3	\$ 91.300	\$ 91.300
1	Asesorías programación	\$ 35.000	\$ 35.000
1	Asesorías programación	\$ 130.000	\$ 130.000
1	Purina	\$ 6.500	\$ 6.500
	<b>TOTAL</b>		\$ 570.889

Fuente: Autores

## 9. CONCLUSIONES Y RECOMENDACIONES

El material seleccionado para la elaboración del prototipo no fue el ideal debido a que se requería de un material que brindará un mejor soporte a la estructura usada como base para las celdas de carga, lo cual afectó la precisión de entrega de alimento del dispensador. Además, no facilitó la realización de las pruebas debido a que el material no fluía de forma constante sobre su superficie.

Se vio la necesidad de implementar la programación de alimento de manera local debido a que muchas veces los usuarios pueden no tener un dispositivo Android durante sus horas de trabajo, lo que dificultaría la programación y configuración del mismo, sumado a lo anterior, se debe tener en cuenta que, si bien el alcance del bluetooth se encuentra entre 5 metros a 10 metros, obstáculos como muros pueden afectar la conexión del Smartphone con el dispensador.

El dispensador está dirigido a equinos, sin embargo, el sistema puede ser aplicado a otro tipo de animales, así mismo, se puede implementar una fuente de energía diferente a la seleccionada.

El dispensador tiene como beneficio su bajo costo, estética y baja complejidad a comparación de otras máquinas encontradas en el mercado, esto es un punto a favor al momento de elegir entre el equipo y otro.

El beneficio que el equipo permita entregar alimento para dos equinos al tiempo, reducir el espacio utilizado y ahorrar costos al no requerir un equipo para cada uno.

## BIBLIOGRAFIA

ADAFRUIT. 4x4 Matrix Keypad. 2021. [En línea] [Consultado el 10 de enero de 2021] Disponible en: <https://www.adafruit.com/product/3844>

ALI, Zahid. Introduction to HC-05. TheEngineeringProjects.com. 20 de octubre de 2019. [En línea] [Consultado el 11 de enero de 2021] Disponible en: <https://www.theengineeringprojects.com/2019/10/introduction-to-hc-05.html>

ALIBABA.COM. Adaptador de alimentación de entrada de 100 de 240v AC 50/60hz 0.6a a DC 6V 6,5 V 100ma 300ma 350ma 450ma 500ma 800ma 1200ma 1500ma AC DC adaptador de potencia. 2021. [En línea] [Consultado el 13 de enero de 2021] Disponible en: <https://spanish.alibaba.com/product-detail/power-adapter-input-100-240v-ac-50-60hz-0-6a-to-dc-6v-6-5v-100ma-300ma-350ma-450ma-500ma-800ma-1200ma-1500ma-ac-dc-power-adapter-60839475759.html>

ARDUINO. Arduino mega 2560 REV3. 2021. [En línea] [Consultado el 10 de enero de 2021] Disponible en: <https://store.arduino.cc/usa/mega-2560-r3>

BAKKER, Benne. How to control a character I2C LCD with Arduino. Makerguides.com. 2021. [En línea] [Consultado el 10 de enero de 2021] Disponible en: <https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/>

COLOMBIA. INTITUTO COLOMBIANO AGROPECUARIO. Resolución 061252 (03, febrero, 2020). Por medio de la cual se establecen los requisitos y el procedimiento para el registro de fabricantes e importadores de alimentos para animales, así como los requisitos y el procedimiento para el registro de alimentos para animales y se dictan otras disposiciones. Bogotá D.C., 2020, No. 12. p. 1-19.

COMPONENTES 101. MG996R Servo Motor. 2020. [En línea] [Consultado el 12 de enero de 2021] Disponible en: <https://components101.com/motors/mg996r-servo-motor-datasheet>

DIOSDADO, Raúl. Display LCD mediante bus I2C/TWI/SPI. Zona Maker, 2018. [En línea] [Consultado el 10 de enero de 2021] Disponible en: <https://www.zonamaker.com/arduino/modulos-sensores-y-shields/display-lcd-mediante-bus-i2c-twi-spi>

FERRETRÓNICA. Placa de Desarrollo Mega 2560 R3 Compatible con Arduino. 2021. [En línea] [Consultado el 10 de enero de 2021]. Disponible en: <https://ferretronica.com/products/placa-de-desarrollo-mega-2560-r3-compatible-con-arduino>

FRANCO GUZMÁN Luis Fernando, GALICIA JIMENÉZ Jesús Rodolfo, OSTRIA VALLE Diana. Desarrollo de un sistema de dosificación automático de alimento para equinos. Tesis de Ingeniería de Control y Automatización. México DF.: Instituto Politécnico Nacional. Escuela Superior de Ingeniería Mecánica y Eléctrica, 2010, p.2. Disponible en: <http://tesis.ipn.mx/bitstream/handle/123456789/9820/3.pdf?sequence=1>

GEEK FACTORY. Tarjeta genérica compatible con Arduino Mega 2560. 2013-2020. [En línea]. [Consultado el 12 de noviembre de 2020]. Disponible en: <https://www.geekfactory.mx/tienda/compatibles-con-arduino/arduino-mega-2560-generico/>

HINOJOSA, Diego Mauricio. Diseño de dispensador automático de alimento para animales. Tesis de ingeniería mecatrónica. Quito.: Universidad Tecnológica Equinoccial. Facultad de ciencias de la ingeniería e industrias, 2017, p.56.

ITALCOL ALIMENTOS CONCENTRADOS. Línea Brío Adultos. [En línea]. [Consultado el 23 febrero 2020]. Disponible en: <https://www.italcol.com/equinos/brio/#Brio-Adultos>

LAST MINUTE INGEENIERS. How HC-SR04 Ultrasonic Sensor Works & Interface It with Arduino. 2021. [En línea] [Consultado el 11 de enero de 2021] Disponible en: <https://lastminuteengineers.com/arduino-sr04-ultrasonic-sensor-tutorial/>

LAST MINUTE ENGINEERS. How Servo Motor Works & Interface It with Arduino. 2021. [En línea] [Consultado el 12 de enero de 2021] Disponible en: <https://lastminuteengineers.com/servo-motor-arduino-tutorial/>

LAST MINUTE ENGINEERS. Interface DS3231 Precision RTC Module with Arduino. 2021. [En línea] [Consultado el 10 de enero de 2021] Disponible en: <https://lastminuteengineers.com/ds3231-rtc-arduino-tutorial/>

SURTIFICAN LA PORTADA. Finca olímpico cinta azul. [En línea]. [Consultado el 23 febrero 2020]. Disponible en: <https://surtifincalaportada.com/45/producto/finca-olimpico-cinta-azul/>

ZALDIVAR PEDROSO, Israel de Jesús. Manual de Equinos. [En línea]. La Habana.: Asociación Cubana de Producción Animal. 2006. [Consultado el 21 febrero 2020]. Disponible en: <https://biblioteca.ihatuey.cu/link/libros/veterinaria/mae.pdf>