



UNIVERSIDAD TECNOLÓGICA ISRAEL

TRABAJO DE TITULACIÓN EN OPCIÓN AL GRADO DE:

INGENIERO EN ELECTRÓNICA DIGITAL Y TELECOMUNICACIONES

**TEMA: DESARROLLAR UNA VÁLVULA DE RIEGO AGRÍCOLA MOTORIZADA
PROGRAMABLE MEDIANTE BLUETOOTH, AUTÓNOMA Y MOVIL.**

AUTOR: JUAN IGNACIO MÉNDEZ GARCÍA

TUTOR: Mg. RENÉ ERNESTO CORTIJO LEYVA.

QUITO- ECUADOR

AÑO: 2018

DECLARACIÓN

Yo, Juan Ignacio Méndez García, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado profesional; y, son responsabilidad del autor.

.....

Juan Ignacio Méndez García

CERTIFICACIÓN DEL TUTOR
UNIVERSIDAD TECNOLÓGICA ISRAEL
APROBACIÓN DEL TUTOR

En mi calidad de tutor del trabajo de titulación certifico:

Que el trabajo de titulación **“DESARROLLOR UNA VALVULA DE RIEGO AGRICOLA MOTORIZADA PROGRAMABLE MEDIANTE BLUETOOTH, AUTONOMA Y MOVIL.”**, presentado por el Sr. Juan Ignacio Méndez García, estudiante de la carrera de Electrónica Digital y Telecomunicaciones, reúne los requisitos y méritos suficientes para ser sometido a la evaluación del Tribunal de Grado, que se designe, para su correspondiente estudio y calificación.

Quito D.M. septiembre del 2018

TUTOR

.....

Ing. Rene Ernesto Cortijo Leyva, Mg

AGRADECIMIENTOS

Deseo expresar, mi profundo agradecimiento a Dios por todas sus bendiciones, a la Universidad Tecnológica Israel, a mis profesores en especial a mi Tutor el Mg, Rene Ernesto Cortijo Leyva, principal colaborador durante todo este proceso, quien con su dirección, conocimiento y enseñanza permitió el desarrollo de este trabajo.

DEDICATORIA

Dedico este Proyecto de Tesis a mi esposa Maricela Auz, por su apoyo incondicional, a mis hijos Diego Andrés y Juan Ignacio por que los amo infinitamente, a mis padres José Méndez y Rosa García que han sabido darme su ejemplo de trabajo y honradez y a mis suegros Jorge Auz Y Mariana Quelal, que de una u otra manera siempre estuvieron pendientes de este proyecto de estudio.

TABLA DE CONTENIDO

Contenido

Antecedentes.....	14
Planteamiento del problema.....	15
Justificación.....	16
Objetivos.....	16
Objetivo general.....	16
Objetivos específicos.....	16
Alcance.....	17
1. CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA.....	21
1.1 Antecedentes.....	21
1.2 Modelos de sistemas de automatización de sistemas de riego.....	21
1.3 Programadores “Rain Dial R”.....	22
1.4 Características principales.....	22
1.5 Programador autónomo “Node Hunter”.....	23
1.6 Programador por <i>Bluetooth</i> de riego “Galcon 9001BT”.....	24
1.7 Elementos constitutivos de la válvula programable.....	25
1.7.1 Módulo Bluetooth.....	26
1.7.2 Sistema de engranajes.....	27
1.7.3 App inventor.....	31
1.7.4 Batería.....	32
1.7.5 Válvula de compuerta de 2 pulgadas.....	32
2. CAPÍTULO II. MARCO METODOLOGICO.....	21
2.1 Introducción.....	21
2.2 Investigación experimental e inductivo.....	21
2.3 Investigación bibliográfica-documental.....	21
2.4 Población y muestra.....	34
2.5 Procesamiento y análisis de datos.....	34
3. CAPÍTULO III PROPUESTA.....	35
3.1 Descripción general del proyecto.....	35
3.2 Diagrama de bloques del sistema.....	36
3.3 Diagrama de flujo.....	37
3.4 Módulos del sistema.....	39

3.5	Aplicación de configuración del sistema.	39
3.6	Módulo de control.	39
3.6.1	Arduino Nano	40
3.7	Actuadores.	42
3.7.1	Servomotor MG-995.	43
3.8	Sensores.	44
3.8.1	Módulo sensor de velocidad FC-03.	44
3.9	Comunicación.	45
3.9.1	Módulo <i>Bluetooth</i> HC-05	46
3.10	Programas.	47
3.10.1	Software FreeCAD.	47
3.10.2	App Inventor.	48
3.10.3	Arduino IDE (Entorno de desarrollo integrado)	48
3.10.4	Proteus.	48
3.11	Aspectos técnicos del producto.	49
3.12	Análisis de costos para el desarrollo del proyecto.	49
3.12.1	Placa de control.	50
3.12.2	Módulo <i>Bluetooth</i> .	51
3.12.3	Sensor infrarrojo de velocidad.	51
3.12.4	Servomotor.	52
3.12.5	Válvula de compuerta.	52
3.12.6	Presupuesto del proyecto.	53
3.13	Análisis de tiempos.	54
3.14	Ventajas del producto.	60
4	CAPÍTULO IV: IMPLEMENTACIÓN.	61
	Diagrama de bloques.	61
4.1	Diseño de la parte mecánica.	63
4.2	Diseño de piezas para el armado de la válvula	64
4.2.1	Diseño de la base inferior de la carcasa.	65
4.2.2	Diseño del tren de engranes.	69
4.2.3	Cálculo del torque para el vástago de la válvula.	69
4.2.4	Cálculo de relación de transmisión.	71
4.3	Diseño del encoder.	76
4.4	Diseño de la base para las baterías.	76
4.5	Diseño base baterías.	76

4.6	Diseño electrónico.....	80
4.7	Dimensionamiento de fuente de alimentación del circuito.	80
4.7.1	Arduino Nano y sensor de velocidad HC-03.....	83
4.7.2	Arduino Nano y Módulo <i>Bluetooth</i> HC-05.....	84
4.7.3	Arduino Nano y servomotor MG-995.....	85
4.8	Diseño de la parte de software.....	86
4.8.1	Diseño de aplicación conexión <i>Bluetooth</i>	86
4.8.2	Conexión <i>Bluetooth</i>	87
4.8.3	Diseño del menú de configuración horas.....	89
4.8.4	Diseño de visualización de parámetros configurados.....	90
4.8.5	Diseño Lista de válvulas configuradas.....	92
4.8.6	Programación del botón y menú Lista de válvulas Configuradas.	93
4.8.7	Diseño de programación de apertura manual.	94
4.9	Implementación.	96
4.9.1	Armado de la válvula.	96
4.9.2	Instalación servomotor MG-995.	98
4.9.3	Instalación tren de engranes completo.	98
4.9.4	Instalación encoder.	99
4.9.5	Instalación Módulo <i>Bluetooth</i>	100
4.9.6	Instalación del banco de baterías.	101
4.9.7	Instalación panel solar.	101
4.9.8	Verificación del armado.....	102
4.9.9	Ensamblaje completo.	102
4.10	Pruebas de funcionamiento.	103
4.11	Análisis y resultados.....	107
	CONCLUSIONES Y RECOMENDACIONES.....	109
	CONCLUSIONES	109
	BIBLIOGRAFIA.....	112
	ANEXOS	113

LISTA DE FIGURAS

Figura. 1.1 Programador de riego Rain Dial R (Irritrol).....	23
Figura. 1.1.2. Programados autónomo Node Hunter.	24
Figura. 1.3. Programador Galcon 9001BT.	25
Figura. 1.4. Bluetooth banda de 2.4.....	26
Figura. 1.5 Tren de engranes.....	27
Figura. 1.6 Esquema de ejes y torque.....	28
Figura. 1.7 .- Cálculo de fuerza de un disco.....	29
Figura. 1.8. Cálculo de Torque.....	30
Figura. 1.9 App inventor 2 herramienta de diseño web.....	31
Figura. 1.10. Batería externa o power bank.....	32
Figura. 1.11. Válvula de compuerta 2”.....	32
Figura. 3.1. Diagrama de conexión de elementos,	36
Figura. 3.2 Diagrama de bloques del sistema.	36
Figura. 3.3. Diagrama de flujo del sistema.....	38
Figura. 3.4. Arduino Nano.....	40
Figura. 3.5. Distribución de pines Arduino Nano.....	42
Figura. 3.6. Servo motor MG-995.....	43
Figura. 3.7. Módulo de velocidad FC-03	45
Figura. 3.8 . Módulo Bluetooth HC-05.....	46
Figura. 4.1. Diagrama de bloques del funcionamiento de la válvula.....	62
Figura. 4.2 Diagrama de Control.....	63
Figura. 4.3. Base de carcasa de la valvular,	64
Figura. 4.4. Válvula de 2 pulgadas.....	65
Figura. 4.5 Diseño 3D base vista superior válvula.....	66
Figura. 4.6 Diseño 3D base vista lateral válvula.....	67
Figura. 4.7 Diseño 3D base vista inferior válvula.....	68
Figura. 4.8 Tren de engranes.....	69
Figura. 4.9 Cálculo torque.....	70
Figura. 4.10 Relación entre engranes.	71
Figura. 4.11 Diseño 3D engrane válvula.....	73
Figura. 4.12 Diseño 3D engrane servomotor.....	74
Figura. 4.13 Diseño 3D base engranes.....	75
Figura. 4.14 Diseño 3D de encoder.	77
Figura. 4.15 Diseño 3D base baterías.....	78
Figura. 4.16 Diseño 3D carcasa superior.....	79
Figura. 4.17 Diagrama circuital.	81
Figura. 4.18 Diseño sensor velocidad HC-03 y Arduino.....	84
Figura. 4.19. Diseño módulo Bluetooth y Arduino,.....	85
Figura. 4.20. Comunicaciones Arduino Nano- servomotor	86
Figura. 4.21 Diagrama de flujo de conexión Bluetooth.....	87
Figura. 4.22. Conexión Bluetooth con válvula,.....	88
Figura. 4.23. Opción de igualar reloj.....	89
Figura. 4.24 Diagrama de flujo de igualar hora,	90
Figura. 4.25. Menú visualización datos para grabar,.....	91
Figura. 4.26 Diagrama de flujo de visualización de parámetros configurados.	92

Figura. 4.27 Diagrama de flujo de visualización lista de válvulas configuradas.	93
Figura. 4.28. Menú mostrar válvulas configuradas.	94
Figura. 4.29 Diagrama de flujo de visualización lista de válvulas configuradas.	95
Figura. 4.30. Apertura manual	96
Figura. 4.31. Base inferior de soporte de válvula.	97
Figura. 4.32 Montaje de la base sobre la válvula.	97
Figura. 4.33. Instalación de servomotor MG_995.	98
Figura. 4.34. Tren de engranes y base.	98
Figura. 4.35. Soportes de base de engranes.	99
Figura. 4.36 Mecanismo de engranes completo.	99
Figura. 4.37 Acople de encoder y sensor infrarrojo de velocidad.	100
Figura. 4.38. Instalación módulo Bluetooth.	100
Figura. 4.39 Instalación banco de baterías y Arduino Nano.	101
Figura. 4.40. Conexión panel solar.	101
Figura. 4.41. Validación con todos sus componentes.	102
Figura. 4.42 Armado complete de la válvula.	102
Figura. 4.43. Pruebas de encendido y apagado de la válvula	103
Figura. 4.44. Conexión Bluetooth entre Smartphone y válvula.	104
Figura. 4.45. Parámetros configurados en la válvula.	105
Figura. 4.46. Parámetros configurados y visualizados en la válvula.	106
Figura. 4.47. Apertura y cierre de la valvula a la hora indicada.	107
Figura. 0.1 Partes internas de la válvula	138
Figura. 0.2 Conexión Bluetooth	139
Figura. 0.3. Verificación de parámetros configurados	140

LISTA DE TABLAS

Tabla 3.1. Características Arduino Nano.....	42
Tabla 3.2 Precios microcontrolador Arduino.....	51
Tabla 3.3 Precios módulo Bluetooth.....	51
Tabla 3.4 Precios Sensor infrarrojo de velocidad.....	52
Tabla 3.5 Precios servomotor.....	52
Tabla 3.6 Precios de válvulas metálicas de compuerta.....	53
Tabla 3.7 Tabla de costos totales.....	54
Tabla 3.8 Cronograma de Actividades.....	56
Tabla 3.9 Cronograma de Actividades.....	57
Tabla 3.10 Cronograma de Actividades.....	58
Tabla 3.11 Cronograma de Actividades.....	59
Tabla 4.1 Consumo de corriente teórica y práctica.....	80
Tabla 4.2 Tabla conexión infrarrojo y Arduino Nano.....	83
Tabla 4.3 Tabla conexión HC-05 y Arduino Nano.....	84
Tabla 4.4 Tabla conexión HC-05 y Arduino Nano.....	85
Tabla 4.5 Análisis de conexión Bluetooth.....	104
Tabla 4.6 Pruebas de envío de configuración.....	105
Tabla 4.7 Pruebas de envío de configuración.....	106

RESUMEN

El desarrollo de la válvula de riego motorizada programable tiene la finalidad de ayudar las labores de riego de una manera automatizada a los pequeños, medianos, grandes agricultores y de igual manera de la industria agrícola.

Para el desarrollo de la válvula fue necesario investigar los tipos de sistemas de riego que existen en la actualidad, de igual forma el funcionamiento de estos. En este caso el prototipo consta de tres partes fundamentales en la elaboración de la válvula que son: la parte electrónica, mecánica y software. La parte mecánica consta del sistema de engranes, soportes, carcasa que fueron diseñados y construidos gracias a la ayuda de la impresión 3D en material PLA. La parte electrónica está constituida por el microcontrolador Arduino Nano, sensor de velocidad, módulo *Bluetooth* y un servomotor. En la parte de software que viene a ser la interface HMI desde el dispositivo *Smartphone* y la configuración del microcontrolador Arduino Nano, el conjunto de estos hace la funcionalidad de la válvula motorizada programable.

El funcionamiento del prototipo de la válvula funciona de la siguiente manera: se dispone de un dispositivo con sistema operativo *Android* que puede ser un *Smartphone* o *Tablet* el cual tiene instalado el programa que tendrá interacción con la válvula, este se conectará mediante *Bluetooth* con la válvula, una vez conectado se procederá a la configuración de esta, la cual consiste en configurar la hora actual, hora de apertura y cierre, así también del armado. Este proceso se lo realiza con cada una de las válvulas que se disponga, como tarea adicional se puede registrar cada configuración realizada en cada válvula con el fin de tener una base de datos de los registros de todas las válvulas que se dispone.

Palabras: Automatizar, prototipo, ayuda, interacción, configuración

Abstract.

The development of the programmable motorized irrigation valve has the purpose of helping the irrigation work in an automated way to the small, medium, large farmers and in the same way of the agricultural industry.

For the development of the valve it was necessary to investigate the types of irrigation systems that exist today, as well as the operation of these. In this case the prototype consists of three fundamental parts in the elaboration of the valve that are: the electronic, mechanical and software part. The mechanical part consists of the gear system, supports, casing that were designed and built thanks to the help of 3D printing in PLA material. The electronic part consists of the Arduino Nano microcontroller, speed sensor, Bluetooth module and a servomotor. In the software part that comes to be the HMI interface from the Smartphone device and the configuration of the Arduino Nano microcontroller, the set of these makes the functionality of the programmable motorized valve.

The operation of the valve prototype works as follows: There is a device with Android operating system that can be a Smartphone or Tablet which has installed the program that will have interaction with the valve, this will connect Using Bluetooth with the valve, once connected will proceed to the configuration of this, which is to set the current time, opening and closing time, as well as the assembly. This process is done with each of the valves that are available, as an additional task you can register each configuration made on each valve in order to have a database of the records of all the valves that are available.

Keywords. Automated, prototype, helping, interaction, configuration.

Descripción de los capítulos

En el capítulo uno, se detalla la información correspondiente a los elementos utilizados en el desarrollo de la válvula como son: microcontrolador Arduino Nano, módulo *Bluetooth*, sensor infrarrojo de velocidad, servomotor, etc.

El capítulo dos habla sobre la metodología de la investigación utilizada en el desarrollo del presente proyecto.

El capítulo tres habla sobre la propuesta presentada para el desarrollo del proyecto.

En el capítulo cuatro se describe el diseño, la implementación, programación, armado y desarrollo del prototipo, pruebas de funcionamiento, y manejo del aplicativo.

Como parte final se mencionan las conclusiones y recomendaciones recopiladas durante la construcción del prototipo, también se citan las fuentes bibliográficas y se adjuntan anexos.

INTRODUCCIÓN

Antecedentes.

Los primeros sistemas de riego son antiguos de los siglos XX, estos sistemas de riego fueron utilizados en el hogar para el riego en césped, plantas ornamentales etc. Luego los sistemas de riego fueron desarrollándose paulatinamente. Los sistemas por aspersión son los más utilizados actualmente, cerca de un 10% de los riegos de cultivos utilizan el sistema por aspersión, este sistema de riego es aplicado a todo cultivo y tipo de suelo. (CREA-UCLM 2010).

La mayoría de estos sistemas de riego por aspersión aún usan tuberías que son movidas de forma manual, por el alto costo y carencia de la mano de obra se ve la necesidad de la automatización de los riegos que es cada vez más frecuente, y viene dada a través de la utilización de válvulas que controlan la apertura, flujo de agua hacia las tuberías y aspersores, una forma más avanzada son las válvulas que son programables en tiempo de acuerdo con la secuencia deseada. Existen sistemas más sofisticados en los cuales se utiliza sistemas de control, que son programados y estos ordenan automáticamente la apertura y cierre de las electroválvulas de acuerdo con el tipo de cultivo y manejo del riego deseado. En definitiva, la automatización de riego en los campos agrícolas permite que se pueda regar varios campos sin la intervención exagerada de la mano de obra, de esta manera se dispone de una unidad de control y varias electroválvulas gestionadas (CREA-UCLM 2010).

Por lo general, un cultivo del tubérculo de papa necesita entre 600 y 700 mm de agua en su producción, distribuida en forma uniforme a lo largo de su periodo de cultivo, la época crítica que se debe dar más atención a los cultivos es en la floración, tuberización. En condiciones normales se dispone de un grado de humedad entre 700 a 800 mm distribuidos en todo el año, el riego se lo realiza por lo general en épocas de sequía, cuando se realizan siembras en época de verano es necesario tener en cuenta la cantidad de riegos que se realizará en todo el tiempo de cultivo. (INIAP 202).

Planteamiento del problema.

Existen productos agrícolas tradicionales que son consumidos en toda época del año, como es el caso del tubérculo de la papa, sin duda alguna uno de los productos más consumidos en los hogares del Ecuador, este producto es cultivado por lo general en la Sierra Ecuatoriana en épocas de invierno y veranos, para el cultivo de este tubérculo se necesita de grandes cantidades de agua para una buena producción.

En la Sierra Ecuatoriana se tiene un promedio de lluvias que están desde 500 – 750 mm de agua por metro cuadrado, esta distribución de lluvias esta desde los meses de enero hasta mayo, la estación de verano está en los meses de junio a octubre, en esta época es necesario la utilización de sistemas de riego por aspersión o por superficie.

En la región de la Sierra Ecuatoriana, el 95 % de las poblaciones que se dedica a la agricultura, utiliza el método por inundación de surcos o riego tradicional, debido a los escasos del agua, la demanda de riego por aspersión crece poco a poco, el cual hace que este sistema sea más eficiente.

Los sistemas de riego agrícolas que son automatizados utilizan electroválvulas de solenoide, estas son conectadas a una unidad de control principal por medio de cableado. Estos sistemas de riego automatizados trabajan con energía eléctrica suministrada por la red pública, y son de difícil implementación ya que por lo general los cultivos agrícolas (papa) se encuentran alejados de las zonas pobladas.

La automatización de las labores de riego por aspersión en campos agrícolas que se encuentran alejadas de las zonas pobladas es difícil o imposible de realizarlas, ya que la mayoría o en su totalidad de los sistemas de riego trabajan con energía de la red eléctrica; además, estos sistemas funcionan con electroválvulas cableadas hacia el panel principal.

Estas labores de riego agrícola son realizadas en su mayoría en corto tiempo o tiempo limitado, ya que en épocas de verano todos los agricultores necesitan regar sus cultivos, por

esta razón es necesario cumplir estas labores en un tiempo estipulado sin pérdida de tiempo, así como también el ahorro de mano de obra.

Justificación.

La agricultura es un pilar fundamental y primordial en la economía de todo un pueblo, por tal razón es necesario que esta área sea reconocida y atendida de la mejor manera para cumplir con sus necesidades de producción, por ende, es necesario buscar una estrategia para ayudar la parte de riego agrícola, el cual debe ser instalado en zonas alejadas de la ciudad donde no se disponga de energía eléctrica, así como también que sea de fácil manejo y sobre todo móvil.

Desde la óptica productiva, es necesario aprovechar al máximo la disponibilidad del agua para desarrollar la actividad de riego en el tiempo establecido, esto conlleva a ser más eficientes en la utilización de mano de obra, recursos, tiempo, esfuerzo, etc.

Objetivos.

Objetivo general.

Desarrollar una válvula de riego agrícola motorizada programable mediante *Bluetooth*, autónoma y móvil.

Objetivos específicos.

- Desarrollar la automatización de la válvula de riego motorizada con microcontrolador Arduino Nano, servomotor, módulo *Bluetooth*.
- Diseñar el sistema mecánico, electrónico, y de comunicaciones de la electroválvula.
- Construir el sistema mecánico el cual será acoplado y adaptado a las características de la válvula.

- Realizar la programación del microcontrolador Arduino Nano, para el manejo y control de todas las partes que componen la válvula motorizada
- Crear una aplicación que trabaje con Smartphone sobre sistema operativo Android, esta permitirá configurar todos los parámetros de la válvula de forma rápida y eficiente.
- Construir la estructura o armazón que será instalada sobre la válvula, en la cual se encontrará todas las partes constitutivas de la válvula motorizada, esta dispondrá de un grado de protección.
- Realizar las pruebas necesarias en la cuales se validará que todos los elementos trabajen de forma correcta y eficiente, así como también el tiempo de trabajo de la batería recargable.

Alcance

La válvula propuesta trabajará con una batería recargable, con un tiempo de trabajo aproximado de 24 horas tiempo suficiente para ejecutar las labores de riego; esta válvula no necesitará que sea cableada hacia un control principal, esta dispondrá de todos los elementos necesarios para ser autónoma y móvil, y será trasladada e instalada en los lugares alejados donde no se disponga de energía eléctrica.

Los agricultores podrán instalar de forma secuencial varios aspersores, cada uno de estos con su respectiva válvula motorizada, se realizará la programación de forma rápida y eficiente de cada una de las válvulas de forma secuencial, se garantizará que la operación de riego no tenga interrupciones y se aproveche al máximo el tiempo de riego.

Se realizará el diseño y la construcción para convertir una válvula normal de agua de 2 pulgadas en una válvula motorizada programable mediante el sistema inalámbrico

Bluetooth, esta será autónoma, y móvil; no necesitará de cables de conexión hacia un sistema principal como los sistemas de riego tradicionales que trabajan con energía eléctrica suministrada por la empresa pública.

Esto permitirá, que se logre automatizar el riego de los campos agrícolas alejados de las zonas pobladas donde no se dispone de red eléctrica, esta válvula trabajará con una batería recargable la cual tendrá una autonomía de trabajo de varias horas, tiempo suficiente para realizar las labores de riego que son aproximadas de 24 horas por turno para cada campo agrícola.

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

1.1 Antecedentes.

Desde la antigüedad, el hombre vio la necesidad de disponer de agua para su consumo y campos agrícolas, la carencia de agua en los cultivos forzó al hombre a ingeniarse y así disponer de este líquido vital, por tal razón, es necesario buscar los medios y formas para proveerse de este líquido en los lugares donde no existe, ya sea por tuberías o acequias, etc.

Las primeras formas de riego aparecieron hace aproximadamente 500 años y consistió en cavar zanjas para transportar el agua a los campos agrícolas.

Con la modernización actual existen varias técnicas de riego agrícola, las más utilizadas actualmente son riego por inundación, aspersión y goteo, en este ámbito el sistema más utilizado es por aspersión para la mayoría de los cultivos agrícolas a campo abierto.

En la actualidad existen varios sistemas de riego automatizados, el cual permite programar la hora y fecha de apertura y de cierre del flujo del agua de riego, estos sistemas por lo general trabajan con electroválvulas solenoides las cuales son conectadas alambreadamente a un equipo de control principal.

1.2 Modelos de sistemas de automatización de sistemas de riego.

En la actualidad existen varios modelos y tipos de sistemas de riego automático los cuales pueden ser implementados en jardines, campos agrícolas e industria.

En el mercado actual los sistemas de riego automáticos se los conoce como *Timers* de riego, estos controlan las electroválvulas que están conectadas directamente por cables.

A continuación, se detallan algunos tipos de electroválvulas programables que existen en el mercado.

1.3 Programadores “Rain Dial R”.

Este es un programador de riego que funciona con una batería de 9V alcalina, para su funcionamiento de programación, este programador trabaja hasta un total de 12 estaciones (electroválvulas), estas electroválvulas son conectadas por medio de cableado.

Es necesario que el programador sea conectado a la red eléctrica, ya que con la pila de 9V no es posible hacer funcionar las electroválvulas, la batería de 9VDC es solo para mantener las configuraciones de lo programado, para el uso de las electroválvulas es necesario más corriente eléctrica.

Este equipo es programable mediante dial y botones, se lo realiza de forma manual para todos los modos de programación.

Este equipo está diseñado para trabajar a la intemperie o en exteriores ya que dispone de una carcasa muy robusta y resistente al clima.

El sistema de riego dispone de la siguiente característica principal que se detallan a continuación, como se aprecian en la figura 1.1.

1.4 Características principales.

- 6, 9 y 12 electroválvulas.
- Módulo electrónico desmontable y programable.
- Tres programas independientes
- Tres tipos de programas.
- Tiempo de riego sin interrupciones: de 1 minuto a 6 horas.

- Repetición semanal de 1 a 15 días o por intervalos.
- Operación manual y semiautomática.
- Pantalla de cristal líquido para visualizar la programación
- Caja robusta con seguridad.
- Conexión por cables.
- Batería alcalina 9 V.
- Programas extras.
- Encendido de válvula.
- Programable con batería.



Figura. 1.1 Programador de riego Rain Dial R (Irritrol).

Fuente: materialesriegos.com

1.5 Programador autónomo “Node Hunter”

Es un programador autónomo de una estación, estable y fiable; sin necesidad de ninguna conexión eléctrica extra para su funcionamiento, basta únicamente de la conexión hacia la batería de 9V.

Este dispositivo está diseñado para presión no muy grandes de un diámetro máximo de 1 pulgada, además de ser resistente a la humedad, agua y evitar la entrada de lodo o tierra.

Características principales:

- Altamente resistente al agua, lodo, polvo, humedad, este puede soportar hasta una profundidad de 3,5 bajo el agua.
- La válvula es de tipo solenoide de accionado por impulsos eléctricos que minimiza su consumo.
- Utiliza una batería de 9VDC.
- Programación fácil por medio de una pantalla LCD.
- Compatibilidad con varios sensores de humedad.
- Se puede realizar hasta un total de nueve programas de riego, con los cuales se puede cumplir las exigencias de todo tipo de riegos cortos, ver figura 1.2.

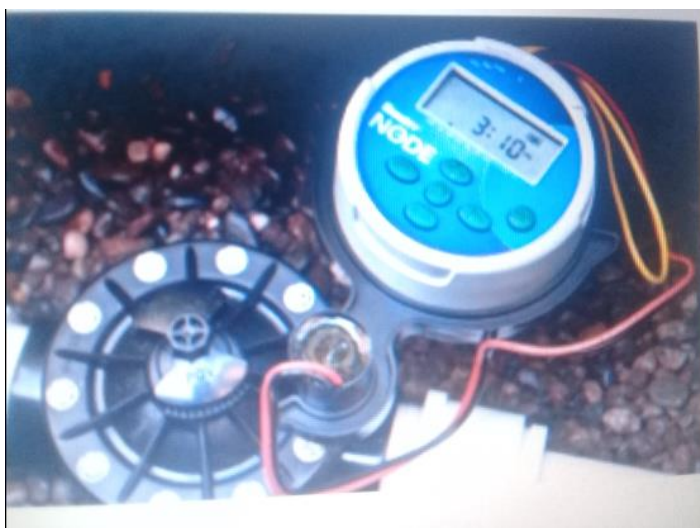


Figura. 1.1.2. Programados autónomo Node Hunter.

Fuente: materialesriegos.com.

1.6 Programador por *Bluetooth* de riego “Galcon 9001BT”

En la figura 1.3 se puede ver un programador de riego, que se utiliza para trabajar de forma manual, con botones o también mediante un dispositivo *Smartphone* vía *Bluetooth*,

con un aplicativo *Android*, este tipo de controlador de riego está diseñado para actividades de riego con baja presión, como riegos de hogar, jardines de casa, etc.

Características principales:

- Resistencia a las inclemencias del clima, de 3/4”.
- Actuador de motor eléctrico.
- Soporta presiones de 1-10 bar.
- Caudal de 50-1200 litros/hora.
- Trabaja una batería de 9VDC.
- Programación semanal o repetitiva.
- Conector opcional para el sensor de humedad.

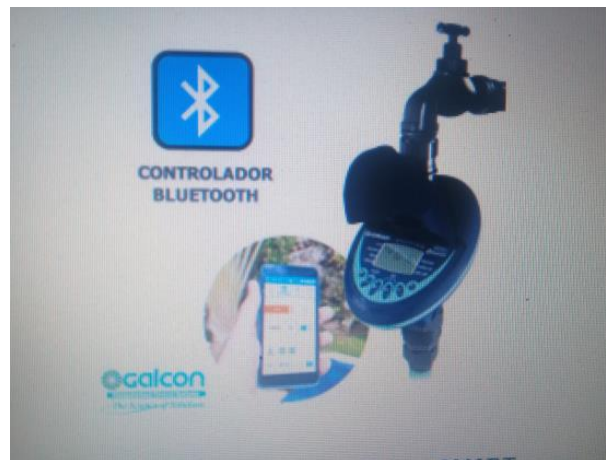


Figura. 1.3. Programador Galcon 9001BT.

Fuente: Elaborado por el Autor.

1.7 Elementos constitutivos de la válvula programable.

A continuación, se mencionarán y se detallarán algunos componentes principales, características que se requieren para el diseño y construcción de una válvula programable.

1.7.1 Módulo Bluetooth

Bluetooth es un tipo de comunicación para redes inalámbricas (*WPAN*) Wireless de áreas personales, este facilita la comunicación entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda ISM de 2,4 GHz.

Este tipo de dispositivos realizan comunicaciones para equipos de bajo consumo de potencia, distancias corta y transceptores de bajo costo.

La comunicación se puede realizar por radiofrecuencia a distancias cortas sin necesidad que los dispositivos se encuentren con línea de vista, según la potencia que manejan, estos dispositivos se clasifican en Clase 1, Clase 2 y Clase 3.

Este depende de la capacidad de canal, los equipos se clasifican en versión 1.2 con un máximo de 1 Mbps, versión 2.0+EDR con 3 Mbps, versión 3.0+HS con 24 Mbps y la versión 4.0 con 32 Mbps.

Para la versión 1.2 tiene una tasa de transmisión bruta de 720 kbps (1 Mbps) con un máximo de 10m de alcance en una frecuencia de 2.4 GHz - 2.48 GHz con la posibilidad de transmisión en Full Dúplex con un total de 1600 saltos por segundo, los saltos se dan en 70 frecuencias a 1 MHz para mayor robustez y confiabilidad, ver figura 1.4.

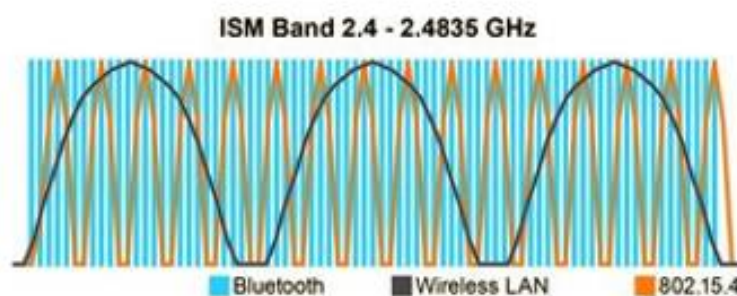


Figura. 1.4. *Bluetooth* banda de 2.4.

Fuente: aprendiendoarduino.wordpress.com

1.7.2 Sistema de engranajes.

Los sistemas mecánicos de engranes permiten transmitir un movimiento giratorio, velocidad, torque de un engrane a otro, se puede modificar la velocidad de giro, sentido y torque, ver figura 1.5.

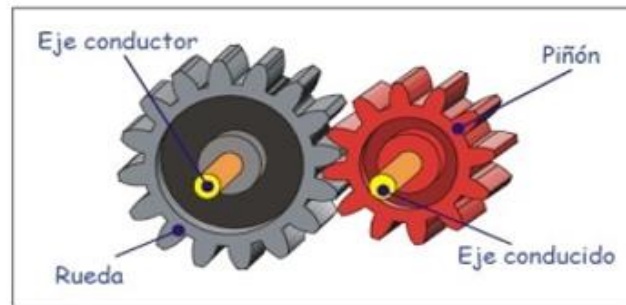


Figura. 1.5 Tren de engranes.

Fuente: Sites google

Un sistema de tren de engranes simples, consta de una rueda motriz con una cantidad de dientes en su periferia exterior, que engrana sobre otros engranes de características similares, lo que evita el deslizamiento entre las ruedas, los engranajes de menor tamaño se llaman piñón y el de mayor diámetro se llama rueda.

➤ Relación de transmisión.

Para el cálculo de la relación de transmisión entre engranes, se necesita disponer de la cantidad de dientes, o también del diámetro de estos, la relación del número de dientes de la rueda conductora con relación al número de dientes de la rueda conducida.

$$\text{Relación de Velocidad} = \frac{\text{Número de dientes de la rueda conductora}}{\text{Número de dientes de la rueda conducida}} = \frac{Z_1}{Z_2} \quad \text{Ec.1}$$

La utilidad de este tipo de mecanismos de engranes es muy amplia, estos pueden ser utilizados en la industria como reductores de velocidad y duplicadores de fuerza, como es el caso de máquinas eléctricas, mecánicas y la mayoría de los electrodomésticos que necesitan reducir la velocidad y aumento de fuerza.

➤ **Cálculo de torque entre engranes.**

En la figura 1.6 muestra el cálculo del torque en un par de engranes, es necesario tener en cuenta que el radio y la fuerza son perpendiculares a los torques del par de engranes.

Para el cálculo del torque se utiliza la siguiente ecuación:

$$T = F \cdot r \text{ (kgf.cm)} \quad \text{Ec.2}$$

donde F es la fuerza y r en el radio de los engranes en centímetros

Las fuerzas $F_1 = F_2$ en el par de engranes son iguales.

F_1 = Fuerza del engrane pequeño

F_2 = Fuerza del engrane grande

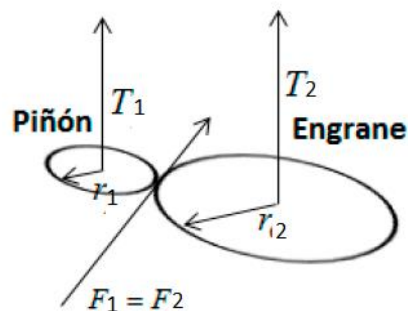


Figura. 1.6 Esquema de ejes y torque.

Fuente scielo.org.mx

$$\begin{aligned}
 F_1 &= F_2 \\
 \frac{T_1}{r_1} &= \frac{T_2}{r_2} \\
 T_2 &= \left(\frac{T_1}{r_1}\right) r_2
 \end{aligned} \quad \text{Ec.3}$$

➤ **Cálculo de fuerza de una polea.**

Para el cálculo de la fuerza para un disco se considera la segunda ley de Newton.

$F = m \cdot g$ (Newtons), donde **m** es la masa en (Kg) y **g** la aceleración de gravedad medida en (m/s^2).

La fuerza resultante es la requerida para mover el disco de un radio **r** con una masa **m** y la aceleración de la gravedad **g** como se muestra en la figura 1.7.

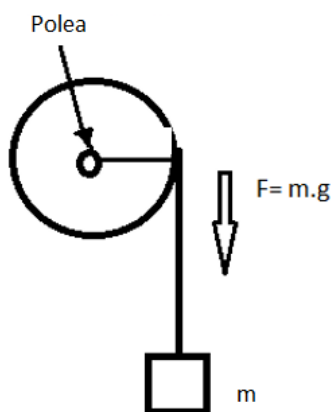


Figura. 1.7 .- Cálculo de fuerza de un disco.

Fuente: Elaborado por el Autor

➤ **Cálculo de torque o momento de fuerza.**

El momento de fuerza o torque es una magnitud vectorial, que mide la capacidad que posee una fuerza para alterar la velocidad de giro de un cuerpo.

$$T = F \cdot r \cdot \text{seno } \theta \text{ (Kgf-cm)} \quad \text{Ec.4}$$

Donde **F** es la fuerza medida en Newtons, y **r** es la distancia del brazo donde se aplica la fuerza en cm, $\text{seno } \theta$ en 90 grados es igual a 1, como se muestra en la figura 1.8.

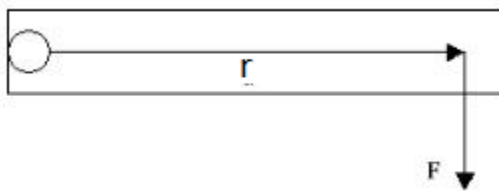


Figura. 1.8. Cálculo de Torque.

Fuente: bachilleratoenlinea.com

➤ **Cálculo de consumo eléctrico.**

Para el cálculo del tiempo de funcionamiento de un circuito eléctrico, se toma en cuenta la capacidad de carga de las baterías, el cual está dado en mA/h.

$$\text{Tiempo de trabajo} = \frac{C}{I} \text{ (horas)} \quad \text{Ec.5}$$

Donde **C** es la capacidad de carga de la batería en Amperios hora, **I** es el consumo de corriente en Amperios.

Para el cálculo de carga de una batería, se tiene que considerar la capacidad de carga de las baterías y del cargador

$$\text{Tiempo de carga} = C \cdot I_c \text{ (horas)} \quad \text{Ec.6}$$

Donde **C** es la capacidad de carga de la batería en Amperios hora, **Ic** es la capacidad de corriente que suministra el adaptador de corriente o cargador.

1.7.3 App inventor.

Es un aplicativo de desarrollo de software, creado por Google para realizar programas de aplicaciones destinadas al sistema operativo *Android*. El programador lo puede desarrollar de forma visual y fácil, a partir de un conjunto de herramientas básicas que pueden desarrollarse en línea. El sistema es gratuito y se puede acceder con una cuenta de Google, que consta con todas las funcionalidades de las herramientas. Las aplicaciones creadas en esta herramienta pueden estar limitadas por su simplicidad, a pesar de este inconveniente estas aplicaciones pueden cubrir una gran cantidad de necesidades básicas para ser utilizadas en dispositivo móvil con sistema operativo *Android*.

En la figura 1.9 se puede apreciar el aplicativo de *App inventor*, en la cual se observa la configuración por bloques, estas se enlazan entre sí para crear una aplicación que será ejecutada en un dispositivo Smartphone con sistema operativo *Android*.



Figura. 1.9 App inventor 2 herramienta de diseño web.

Fuente: página de ai2.appinventor.mit.edu

1.7.4 Batería.

Las baterías externas tienen diferentes capacidades de carga, estas pueden ser medidas en mAh (miliamperio x hora), esta unidad de medida es utilizada para medir la capacidad de carga eléctrica acumulada en un determinado período de tiempo, mientras mayor sea la capacidad de carga, esta podrá suministrar energía por más tiempo a los dispositivos conectados. En la figura 1.10 se puede observar un tipo de batería recargable o Power Bank.



Figura. 1.10. Batería externa o power bank.

Fuente: ebay.com

1.7.5 Válvula de compuerta de 2 pulgadas

En la figura 1.11 se puede observar una válvula de compuerta, está compuesta por el levantamiento de una compuerta y por una cuchilla, construida de latón y cobre, permite el paso de fluidos, puede soportar presiones superiores a 125 PSI.



Figura. 1.11. Válvula de compuerta 2”.

Fuente: www.mi-fontaneria.es

CAPÍTULO II. MARCO METODOLOGICO

2.1 Introducción.

En el presente proyecto se aplicó la investigación aplicada o empírica, la cual se refiere al estudio científico que permite resolver problemas prácticos, con la cual se busca la aplicación de la utilización de los todos los conocimientos técnicos y científicos, adquiridos durante el proceso de toda la formación académica y práctica, además la información adquirida en el proceso formativo sobre la tecnología actual.

2.2 Investigación experimental e inductivo.

Se utilizó el método experimental e inductivo, para el desarrollo del tipo de proyecto seleccionado debido a que fue necesario acoplar módulos de electrónica, mecánica, comunicaciones, programación en microcontroladores y dispositivos Smartphone, con lo cual se consiguió obtener los resultados deseados, y tener la interacción de la interface HMI con la plataforma construida.

2.3 Investigación bibliográfica-documental

El proyecto de investigación se desarrolló con una modalidad bibliográfica, ya que en primera instancia se revisó proyectos de tesis de grado, que tiene alguna semejanza con el tema propuesto, además de tener un enfoque más amplio del tema seleccionado el desarrollo de este.

Esta investigación se fundamentó básicamente en información recopilada de internet, folletos, revistas, y alguna otra información sobre el tema, enfocándose en los métodos y equipos utilizados para el desarrollo del riego agrícola.

2.4 Población y muestra.

Para el desarrollo del proyecto no fue necesario desarrollar las muestras de población, estadísticas, etc. ya que el proyecto se centra más en lo experimental.

2.5 Procesamiento y análisis de datos

Con la recopilación de información relevante para el proyecto de investigación se desarrollarán los siguientes pasos para establecer la mejor manera del desarrollo de este.

Se procederá a la organizar de la información obtenida en la investigación.

De manera minuciosa se revisará la información.

Se procederá a un análisis minucioso de la información recopilada.

Encontrar la solución al problema planteado.

Interpretar de mejor manera los resultados obtenidos.

CAPÍTULO III PROPUESTA

3.1 Descripción general del proyecto.

El desarrollo de la construcción de una válvula motorizada constará de modo general de los siguientes elementos:

Se dispondrá de una válvula de compuerta metálica de 2 pulgadas la misma que será modificada para convertirla en una válvula motorizada programable.

La válvula estará compuesta de varios elementos electrónicos, mecánico y de comunicaciones que trabajarán de forma integrada, para convertir una válvula común en una válvula motorizada y programable.

En la figura 3.1, se detalla un diagrama de los elementos constitutivos de la válvula, además, de la interacción de cada una de estas partes.

Esta válvula será controlada por un dispositivo *Smartphone* con sistema operativo *Android*, que tendrá el aplicativo realizado en **App inventor**, este constará de todas las instrucciones necesarias para configurar la válvula, entre las principales configuraciones que puede realizar son: configuración de hora actual, hora de apertura, hora de cierre de la válvula, armado de la válvula, también se dispone de una apertura y cierre de forma manual en caso de presentarse algún problema, y de una base de datos de todas las válvulas que se configuraron con sus respectivos parámetros.

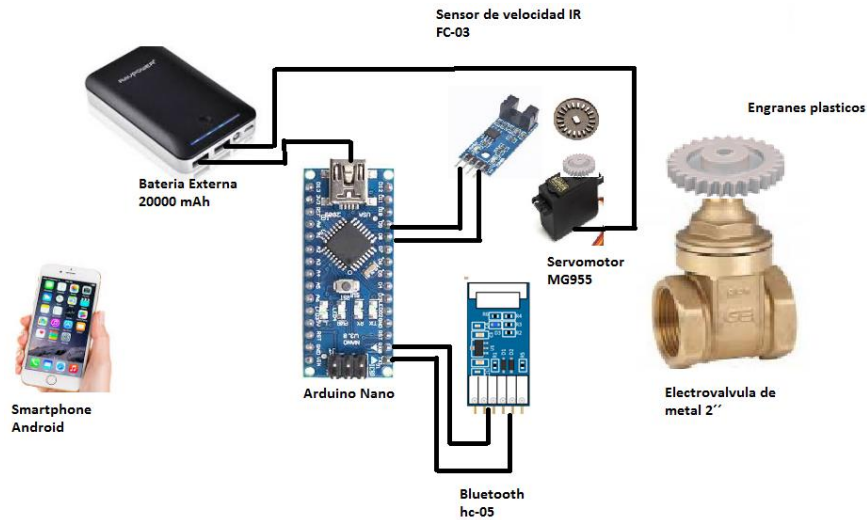


Figura. 3.1. Diagrama de conexión de elementos,

Fuente: Elaborado por el autor.

3.2 Diagrama de bloques del sistema.

Para el presente proyecto se realiza el diagrama de bloques, este consta de los principales elementos del circuito y la comunicación, como se muestra en la figura 3.2.

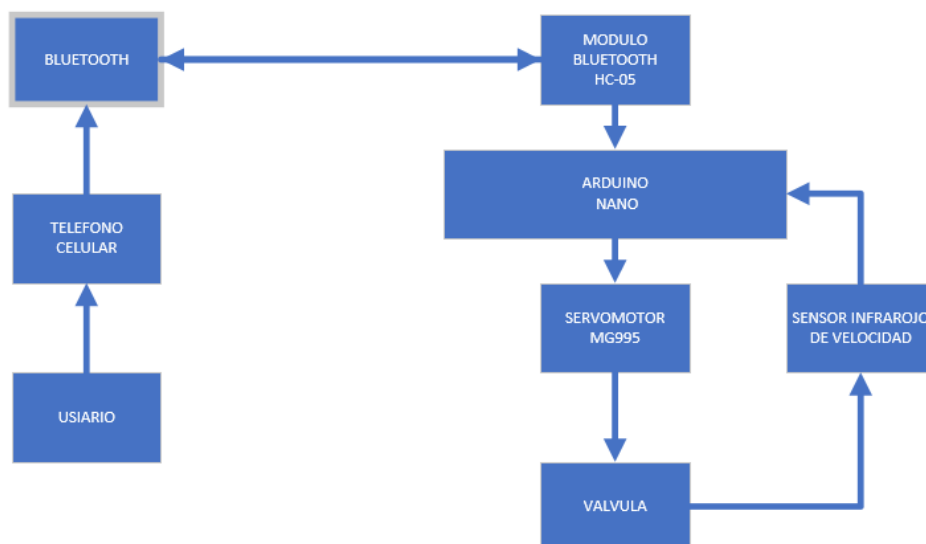


Figura. 3.2 Diagrama de bloques del sistema.

Fuente: Elaborado por el autor.

3.3 Diagrama de flujo.

El diagrama de flujo de la figura 3.3, especifica el modo de comunicación de todo el sistema, empieza con la inicialización de la aplicación del *Smartphone*, que contiene todas las opciones necesarias para realizar las configuraciones de la válvula, se conecta por *Bluetooth* con el módulo HC-05, y trabaja en modo *Slave*, es una conexión uno a uno con el dispositivo *Smartphone*.

Una vez realizado en emparejamiento o conexión *Bluetooth*, se prosigue con la configuración de la válvula, el primer parámetro a configurar es la hora actual, esta es la misma hora que dispone el *Smartphone*, luego se configura la hora de apertura y cierre de la válvula, estos parámetros son ingresados en el Arduino NANO.

Internamente el microcontrolador Arduino NANO, realiza la comparación de la hora actual con la hora de apertura y cierre de la válvula, si estos valores son iguales se ejecuta la orden de encender el servomotor para cerrar o abrir la válvula.

El sensor de velocidad censa el movimiento del vástago de la válvula, mientras existe movimiento del vástago el sensor envía una señal inicial que siga girando el servomotor, si no existe movimiento del vástago el sensor de velocidad envía una señal que indica que se pare el servomotor, esta operación la realiza al momento de abrir o cerrar la válvula.

De igual manera, el usuario puede verificar los parámetros que fueron configurados en la válvula, esto es factible ya que la aplicación tiene la opción de envío y recepción de datos, esta opción ayudará a comprobar que los parámetros ingresados se encuentren correctamente configurados.

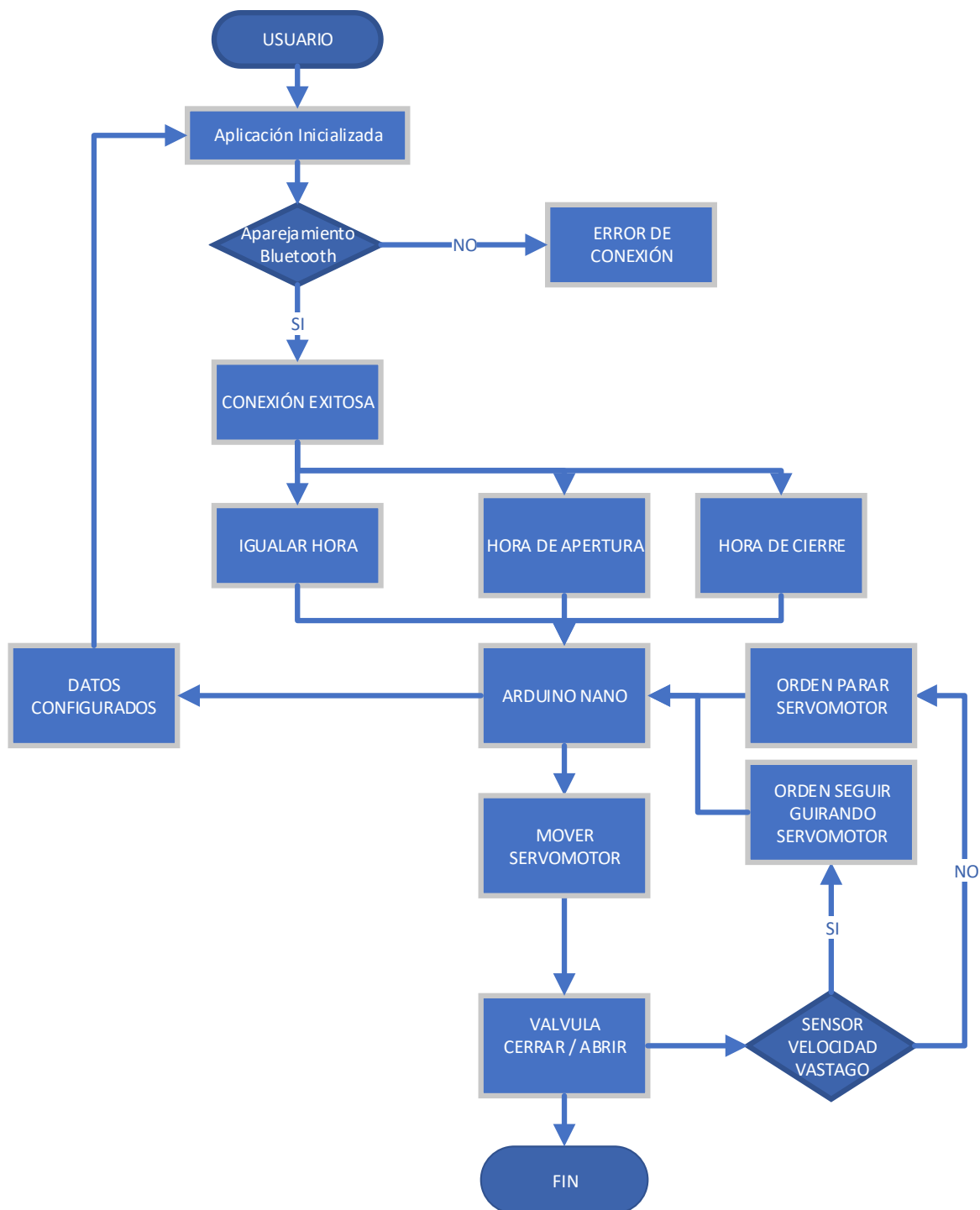


Figura. 3.3. Diagrama de flujo del sistema.

Fuente: Elaborado por el Autor

3.4 Módulos del sistema.

El sistema completo que comprende la válvula motorizada se subdivide en módulos, cada uno de estos consta con los elementos y dispositivos electrónicos que lo conforman, estos son descritos como parte de la propuesta.

Estos módulos son:

- Aplicación de configuración del sistema.
- Módulo de control.
- Actuadores.
- Sensores.
- Comunicación.

3.5 Aplicación de configuración del sistema.

La aplicación de configuración del sistema es el aplicativo que se encuentra instalado en el dispositivo *Smartphone*, este permite ingresar los parámetros de hora actual, hora de apertura y cierre de la válvula, además de verificar los parámetros que se encuentran configurados actualmente. Los requerimientos necesarios para la instalación del aplicativo son los siguientes.

- Dispositivos Smartphone, con sistema operativo Android con versión 2.3. Giangerbread o superior.
- Smartphone con espacio en memoria de almacenamiento de 15 Megabytes.
- Marcas y modelos de los dispositivos Smartphone, no es un requerimiento para la instalación del aplicativo.

3.6 Módulo de control.

El módulo de control principal es el microcontrolador Arduino Nano, encargado de ejecutar todas las instrucciones que son enviadas desde la aplicación comandada por el

usuario, éste permite administrar y controlar todas las acciones necesarias para el funcionamiento del dispositivo.

3.6.1 Arduino Nano

Arduino Nano es un pequeño controlador, completo y fácil de usar basado en *ATmega328* (Arduino Nano 3.0), este es semejante en funcionalidades que el Arduino Uno, pero de diferente tamaño. Este no dispone de una alimentación para voltaje DC, trabaja con un puerto Mini-b USB. En la figura 3.4 se puede apreciar la distribución de pines del microcontrolador Arduino Nano.

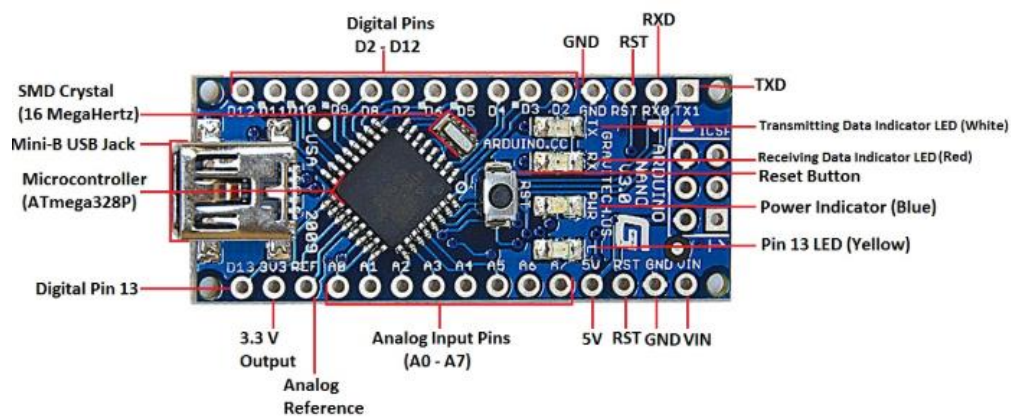


Figura. 3.4. Arduino Nano.

Fuente: store.arduino.cc

3.6.1.1 Fuente de poder

Arduino Nano, puede ser encendido con un puerto Mini B USB, además de soportar entre 6-12 V DC con una fuente DC externa por el pin 30.

3.6.1.2 Memoria.

El chip *ATmega 328* tiene 32 KB, de los cuales 2 KB son usados para el arranque del *bootloader*, además dispone de 2 KB de memoria RAM y 1 KB de memoria EEPROM.

3.6.1.3 Entradas y salidas.

Dispone de 14 pines digitales que son usados como pines de entradas o salida, estos operan con 5 VDC, cada pin puede recibir o suministrar un máximo de 40 mA, además de disponer algunos pines extras que tienen funciones especiales.

Arduino Nano dispone de 8 entradas analógicas con una resolución de 10 bits, por defecto de 0 a 5 voltios.

3.6.1.4 Comunicación.

Arduino Nano tiene la facilidad de comunicarse con un computador por medio de un puerto serial, el *ATmega328* provee *UART TTL (5V)*, además dispone de puertos digitales (*RX*) (*TX*) para transmisión inalámbrica, el software de Arduino provee un puerto virtual con la computadora para recibir y enviar datos en texto plano desde el Arduino y el computador.

3.6.1.5 Programación.

Los microcontroladores Arduino, son programados con el software de Arduino que es obtenido desde la página web del fabricante, este software es totalmente gratuito.

Es un entorno de desarrollo abierto, que permite programar un dispositivo que sea compatible a este lenguaje, se base en el lenguaje C, tiene predefinido librerías que permiten mejorar las líneas de programación en el proceso de construcción.

3.6.1.6 Características Arduino Nano.

En la tabla 3.1 se mencionan las principales características del microcontrolador Arduino Nano, en la figura 3.5 se puede ver la distribución de pines del Arduino Nano.

<i>Características Arduino Nano</i>	
Microcontrolador:	Atmega 328
Voltaje de alimentación de entrada:	5V
Pines digitales:	14 (6 con PWM)
Puertos analógicos:	16
Corriente máxima suministrada por pin:	40mA
Corriente máxima suministrada para el pin 3.3V:	50 mA
Memoria flash:	32 kB
Memoria SRAM:	8 kB
Memoria EEPROM:	2 kB
Velocidad de reloj:	16 MHz

Tabla 3.1. Características Arduino Nano.

Fuente: store.arduino.cc

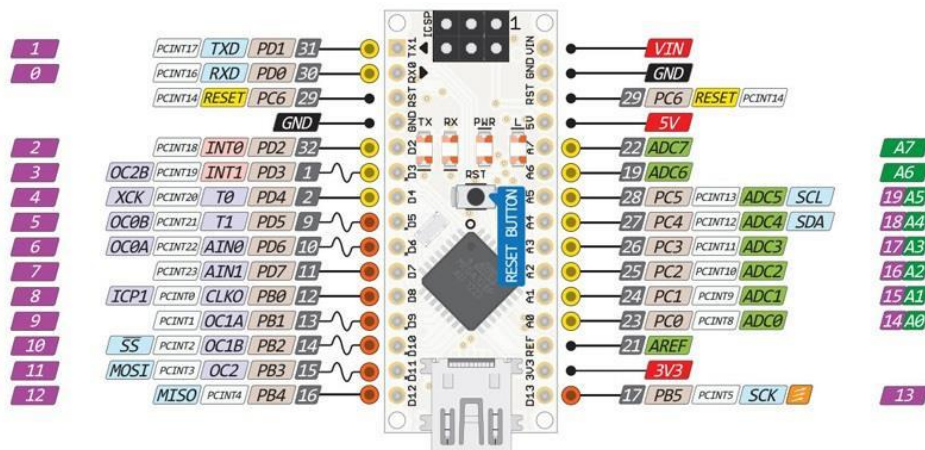


Figura. 3.5. Distribución de pines Arduino Nano.

Fuente: store.arduino.cc

3.7 Actuadores.

Son los dispositivos que transforman la energía eléctrica en un proceso automatizado, mediante la administración de sus movimientos con el microcontrolador Arduino Nano, el

actuador que se utiliza es el servomotor MG-995, este estará encargado de realizar la apertura y cierre de la válvula.

3.7.1 Servomotor MG-995.

Los servomotores son motores electrónicos, que mueven o empujan partes de máquinas con gran precisión, los servomotores se los encuentra en artefactos electrónicos, juguetes, etc.

La simplicidad de los servomotores los hace muy confiables y robustos, la parte principal del servomotor es un pequeño motor que funciona con corriente continua, el cual gira a grandes velocidades expresadas en (RPM) pero de muy poco torque, un arreglo de engranes hacen que la gran velocidad del motor y bajo torque sea convertido en baja velocidad y un alto torque.

El Servomotor MG-995 dispone de una modulación digital, además con un gran torque comprendido entre 8,5 kg-cm con un voltaje de 4,8 VDC hasta un máximo de 10 kg-cm con 6 VDC, también dispone de un sistema de engranes de metal el cual está diseñado para trabajo continuo y de gran torque, como se observa en la figura 3.6.

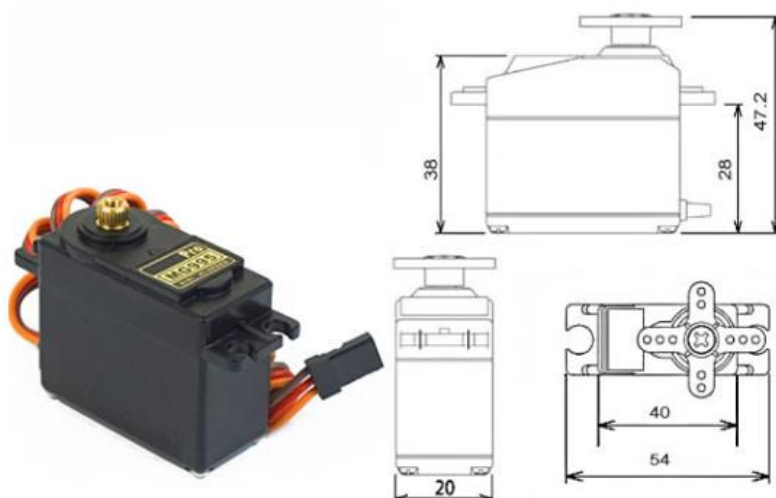


Figura. 3.6. Servo motor MG-995.

Fuente: electronicoscaldas.com

Principales características:

- Modelo: MG-995
- Tamaño: 40.7, 19.7, 42.9mm
- Peso: 55g
- Torque 8,5 Kgf-cm / (4,8V), 10 Kgf-cm (6V)
- Velocidad: 53-62 RPM.
- Temperatura de operación -30 hasta 60 °
- Tipo de conector: JR, FUTABA General
- Angulo de rotación: 180 grados (truncado para giro continuo).
- Tipo: Servo
- Corriente de operación: 100mA.

3.8 Sensores.

Son dispositivos que tienen la propiedad sensible a una magnitud del medio, y transformarlas con un transductor en una variable eléctrica. El sensor que transforma la propiedad del movimiento en una variable eléctrica es el sensor infrarrojo de velocidad FC-03, este censa el movimiento del vástago de la válvula en una variable eléctrica.

3.8.1 Módulo sensor de velocidad FC-03.

El sensor de velocidad está compuesto de un transmisor y receptor por infrarrojos, además de un encoder ranurado, este genera una serie de impulsos digitales que son utilizados como sensor de velocidad con la programación adecuada.

Los codificadores ópticos, están constituidos por un disco transparente o discos ranurados con una serie de marcas opacas colocadas radialmente y equidistantes, el sensor consta de un LED infrarrojo y un receptor los cuales detectan el paso de luz que atraviesa el encoder ranurado, estos pulsos son detectados y contabilizados, se puede observar el sensor en la figura 3.7.



Figura. 3.7. Módulo de velocidad FC-03

Fuente: aprendiendoarduino.wordpress.com

Principales características.

- Voltaje de trabajo: 3,3 a 12 VDC
- Chip principal: LM393
- Salida efectiva: Nivel bajo
- LEDs indicadores de funcionamiento
- Salida: Conmutación digital (0 y 1)
- Cuenta y detecta la velocidad de un motor
- Número de pines: 4
- Compatible con Arduino
- Dimensiones: 31 mm X 14.5 mm
- Modelo: FC-03

3.9 Comunicación.

La tecnología inalámbrica por *Bluetooth* es de corto alcance que permite la comunicación inalámbrica de datos entre dispositivos, el módulo *Bluetooth* a utilizar es el HC-05, este trabaja en modo *SLAVE* por defecto.

3.9.1 Módulo *Bluetooth* HC-05

El módulo *Bluetooth* HC-05 es utilizado en todo tipo de proyectos donde se necesite una conexión inalámbrica fiable, sencilla, económica y de corto alcance. Estos módulos se configuran mediante comandos AT y se tiene la posibilidad de hacerlos funcionar en modo *Máster* o *Slave*, por defecto trabaja en modo *Slave* el cual permite conexiones entrantes.

El módulo *Bluetooth* HC-05 puede ser energizado con un voltaje de 3.3 y 6V, (por defecto con voltaje de 5V), los pines *TX* y *RX* utilizan niveles de voltaje de 3,3V por lo que no se puede conectar directamente a placas Arduino de 5V, el modelo HC-05 es un módulo *Bluetooth* muy versátil, económico y fácil de usar como se puede apreciar en la figura 3.8.



Figura. 3.8 . Módulo *Bluetooth* HC-05.

Fuente: aprendiendoarduino.wordpress.com

Principales características:

- Protocolo de comunicación *Bluetooth*: v1.1 / 2.0.
- Frecuencia de trabajo: banda *ISM* de 2,4 GHz.
- Modulación de trabajo: *GFSK*
- La potencia de transmisión es de menos de 4dBm, Clase 2.

- La sensibilidad es de: Menos de -84dBm en el 0,1% BER.
- Velocidad de transmisión asíncrona de: 2.1Mbps (Max) / 160 kbps.
- Síncrono: 1Mbps / 1Mbps.
- Modo de trabajo: *Bluetooth* (maestro y esclavo).
- Fuente de alimentación DC: + 3.3VDC 50mA. (soporta de 3.3 a 6VDC)
- Temperatura de trabajo en el rango de: -5 ° C a 45 ° C.

3.10 Programas.

En la programación del microcontrolador Arduino Nano, el diseño e impresión de las piezas del prototipo de la válvula y la creación de la aplicación para Android, es necesaria la utilización de programas específicos de cada uno de estos diseños, para realizar esta tarea se escogió los siguientes programas por ser *Free* o por su simplicidad en su manejo.

3.10.1 Software FreeCAD.

Es un programa utilizado para diseñar cualquier tipo objeto sólido, con el objetivo de ser modelado de manera real y si es necesario imprimir o construir según lo amerite. FreeCAD es de diseño CAD 3D (Diseño Asistido por Computador en 3D), tiene una interfaz sencilla, intuitiva y sobre todo no necesita licenciamiento.

- Tiene herramientas para modelación de pieza mecánicas, y no mecánicas.
- Se puede generar planos en formato de 2D, 3D.
- Dispone de herramientas de simulación de piezas.
- Es una herramienta de acceso libre.

El programa FreeCAD, se utiliza con el fin del diseño de las piezas 3D a ser utilizadas en la construcción de la válvula, como es engranes, carcasa, soportes, base, etc. para luego ser impresa en 3D, el material que se utiliza en la impresión es un filamento de plástico llamado PLA (ácido poliláctico o poliláctico), es el más utilizado por su costo, durabilidad y por ser amable con el ambiente.

3.10.2 App Inventor.

App Inventor es un ambiente de desarrollo, que crea aplicaciones para dispositivos móviles con sistema operativo *Android*, éste tiene un ambiente de desarrollo en línea, es decir, no es necesario que sea instalado en un PC, para su programación es necesario disponer de conexión a internet y de un navegador Web.

Esta herramienta de programación se basa en una de línea de desarrollo visual, por bloques de construcción, tiene dos ventanas de trabajo, la primera es el diseño de ventana, en la cual se construye la pantalla de presentación para el usuario con todos los requisitos de ingreso de botones, imágenes, videos, conectividad, etc.

La segunda ventana de trabajo es la de bloques, en esta se ingresan instrucciones por bloques para la ejecución de la aplicación, además de realizar las instrucciones de conectividad con otros dispositivos.

3.10.3 Arduino IDE (Entorno de desarrollo integrado)

Es un entorno de desarrollo abierto que permite programar un dispositivo que sea compatible a este lenguaje, está basado en lenguaje C, este tiene la factibilidad de descargar librerías predefinidas y de crear librerías específicas.

Este tipo de plataforma permite desarrollar una infinidad de programas, los cuales pueden ser ejecutados en cualquier plataforma Arduino.

3.10.4 Proteus.

Proteus es una herramienta de simulación de circuitos electrónicos, el cual dispone de librerías de elementos electrónicos, entre sus principales ventajas está el diseño esquemático electrónico, la programación de ciertos controladores, la simulación del circuito con todos los elementos que lo conforman, con el fin de exportar en un diseño para la construcción del circuito impreso.

El programa Proteus fue utilizado para la elaboración del diagrama electrónico del proyecto, en función de sus elementos electrónicos constitutivos.

3.11 Aspectos técnicos del producto.

El prototipo dispondrá de una conexión inalámbrica Bluetooth, estará conectado mediante un módulo HC-05 en modo esclavo y el dispositivo *Smartphone*, esta conexión será uno a uno para envío y recepción de datos a una distancia no mayor a 10 metros.

Se emplea una válvula metálica de compuerta la cual soporta grandes presiones de caudal, además de la apertura y cierre de forma gradual y no de forma abrupta como la mayoría de los sistemas de riego automatizados, que funcionan con electroválvulas de solenoide, el prototipo utiliza un servomotor y un tren de engranes para ejecutar la apertura y cierre de la válvula para disminuir la velocidad y aumentar el torque sobre el vástago de la válvula.

El prototipo dispone de un sensor infrarrojo de velocidad, este trabaja como un interruptor de fin de carrera, encargado de controlar el funcionamiento del servomotor, garantizando que el servo no gire cuando la válvula se encuentre completamente abierta o cerrada.

Con el aplicativo instado en un dispositivo *Smartphone* se puede configurar varias válvulas, además de llevar un registro de los parámetros configurados en cada una de estas, con el fin de disponer de una base de datos de cada una de las válvulas, con los parámetros configurados.

3.12 Análisis de costos para el desarrollo del proyecto.

Para el análisis de costos, es necesario tomar en cuenta los objetivos planteados en el desarrollo del proyecto, esta servirá de base para investigar los elementos que se acoplen de forma idónea a las necesidades del proyecto, es necesario cumplir con las características y

funciones requeridas y la disponibilidad en el mercado de los elementos constitutivos para su desarrollo.

3.12.1 Placa de control.

Entre las opciones de placa de control, se dispone de la placa del microcontrolador Arduino, este permite la programación de IDE en Arduino, además cuenta con las especificaciones del diseño.

Dentro de las opciones del mercado electrónico se encuentra las placas de Arduino UNO y Nano, las cuales son similares en característica en puertos digitales, analógicos, memoria, etc.

Como primera opción, es escoger el microcontrolador Arduino Uno, este dispone de las características necesarias para desarrollar el proyecto, cuenta con puertos PWM, puertos Tx-Rx, puertos digitales y la opción de alimentación directa, pero se descarta esta opción por el tamaño de la tarjeta, ya que se busca una placa pequeña y compacta, porque el espacio que se dispone es reducido.

Como se segunda opción, es escoger el microcontrolador Arduino Nano, el cual tiene las mismas características que el Arduino UNO, con la diferencia de su tamaño, este se acopla me mejor manera en el espacio reducido para su instalación.

La mejor opción luego del análisis es seleccionar el microcontrolador Arduino Nano, este dispone de las características de puertos PWM, puertos digitales, analógicos, etc. con la ventaja de su tamaño y de su consumo de corriente que es menor que el microcontrolador Arduino UNO, además de los precios similares.

En la tabla 3.2, se presenta una comparación de los precios de los microcontroladores Arduino Nano y Arduino UNO.

COMPARACIÓN DE PRECIOS			
	Arduino UNO	Arduino NANO	
Valor	\$15	\$15	

Tabla 3.2 Precios microcontrolador Arduino.

Se selecciona el microcontrolador Arduino Nano, ya que tiene las mismas funcionalidades del Arduino Nano, además de su tamaño pequeño y su consumo de corriente menor.

3.12.2 Módulo Bluetooth.

Existen dos módulos Bluetooth, que son idóneos para el proyecto, estos cumplen las especificaciones y son de precios similares, estos módulos son de características similares.

En la tabla 3.3, se presentan la comparación de precios de los módulos Bluetooth HC-05 y HC-06.

COMPARACIÓN DE PRECIOS			
	HC-05	HC-06	
Valor	\$10	\$10	

Tabla 3.3 Precios módulo Bluetooth

Se selecciona el módulo Bluetooth HC-05, este puede trabajar en dos modos de operación, además que dispone de las mismas características y precios similares con el módulo HC-06.

3.12.3 Sensor infrarrojo de velocidad.

En el mercado se dispone de vario sensores infrarrojos de velocidad, los cuales cumplen con las características y necesidades deseadas, además de sus precios similares.

En la tabla 3.4, se presentan la comparación de precios de los sensores infrarrojos de velocidad HC-020K y HC-03.

COMPARACIÓN DE PRECIOS			
	HC-020K	HC-03	
Valor	\$7	\$6	

Tabla 3.4 Precios Sensor infrarrojo de velocidad

Se escoge el módulo HC-03 el cual dispone de entrada digital y analógica, con un precio menor, este módulo viene con encoger y cables de conexión.

3.12.4 Servomotor.

Existen dos modelos de servomotores, que cumplen con las características necesarias para el proyecto, estos servomotores son similares físicamente y también sus características internas, se dispone de los modelos MG-995 y el modelo MG-996R.

En la tabla 3.5, se presentan la comparación de precios de los servomotores MG-995 y MG-996R.

COMPARACIÓN DE PRECIOS			
	MG-995	MG-996R	
Valor	\$15	\$17	

Tabla 3.5 Precios servomotor

Se escoge el servomotor MG-995, el cual tiene las mismas características que el otro servomotor, además que el precio es menor.

3.12.5 Válvula de compuerta.

En el mercado existen vario modelos, marcar, características, precios, procedencia, etc. de válvulas metálicas de compuerta en 2 pulgadas.

En la tabla 3.6, se presentan la comparación de precios de dos válvulas de marca REDWHITE de fabricantes diferentes.

COMPARACIÓN DE PRECIOS			
	REDWHITE BRASILERA	REDWHITE ESPAÑOLA	
Valor	\$100	\$120	

Tabla 3.6 Precios de válvulas metálicas de compuerta

Se escoge la válvula RED-WHITE de procedencia brasilera, ya que es de menor precio y de características similares con otras válvulas.

3.12.6 Presupuesto del proyecto.

Luego de haber seleccionado los elementos de todo el circuito, se procede a realizar el presupuesto del proyecto, para el prototipo de la válvula motorizada. En la tabla 3.7 se presenta el presupuesto total del proyecto.

En el análisis de costos se considera el valor involucrado en el desarrollo del proyecto, los elementos constitutivos en la elaboración de la electroválvula motorizada.

Fueron considerados varios puntos de vista, como es el espacio en el cual va a ser instalados, versatilidad, complejidad, calidad, durabilidad, precio, etc.

De igual forma, se consideró la disponibilidad de los elementos que existan en el mercado local, así como también el costo de cada uno de los elementos a ser utilizados en la implementación.

Se desarrolla una tabla en la cual está desglosado el costo, cantidad de cada uno de los elementos constitutivos de la electroválvula.

COSTO TOTAL DEL PROYECTO			
Cantidad	Elemento o Material	Valor Unitario (\$)	Valor total (\$)
1	Módulo Arduino Nano	15	15
1	Módulo <i>Bluetooth</i> HC-05	10	10
1	Sensor de velocidad FC-03	6	6
5	Baterías de 5V 1500mAh	5	25
1	Módulo controlador de carga de Baterías	5	5
1	Panel Solar	6	6
1	Válvula de compuerta de 2 pulgadas RED -WHITE	100	100
1	<i>Encoder</i> de 5 cm de diámetro, 3D	5	5
1	Servomotor MG995 de giro continuo	15	15
1	Diseño y construcción de engranes, 3D	20	20
1	Diseño y construcción de base principal de carcasa, 3D	40	40
1	Diseño y construcción de base de engranes y soportes, 3D	20	20
1	Diseño y construcción de carcasa superior, 3D	40	40
1	Paquete de cables de conexión Arduino, de 10 cm Hembra-Hembra	5	5
	Total		312

Tabla 3.7 Tabla de costos totales

3.13 Análisis de tiempos.

En el análisis de tiempo se realiza una descripción del cronograma realizado durante las fases que comprenden el desarrollo completo del proyecto, en la primera etapa se define los elementos constitutivos de la válvula motorizada, como son elementos electrónicos, mecánicos, software a utilizar, etc. que cumplen con las especificaciones técnicas requeridas, se selecciona los elementos de acuerdo con el costo beneficio para el proyecto. Se tomó 6 días, del sábado 28 de abril al miércoles 5 de mayo del 2018.

La siguiente etapa está dedicada al diseño del sistema mecánico, electrónico y de comunicación, esta tarea tomó 8 días desde el domingo 6 de mayo hasta el jueves 17 de mayo.

En la siguiente tarea está dedicada a la programación del microcontrolador Arduino Nano, se configura cada una de las rutinas para la funcionalidad de esta, se configura el reloj interno, la comunicación con el módulo Bluetooth, sensor de velocidad, servomotor, etc., En realizar esta tarea tomó 22 días desde el viernes 18 de mayo hasta el sábado 16 de junio del 2018.

Se tiene una tarea de configuración de la aplicación para el dispositivo Smartphone, con sistema operativo Android, en realizar el aplicativo tomó 10 días desde el sábado 16 de junio hasta el viernes 29 de junio del 2018.

La tarea de construcción del sistema mecánico, el cual será acoplado y adaptado a las características de la válvula tomó 13 días desde el lunes 2 de julio hasta el miércoles 18 de julio del 2018.

En la siguiente tarea se tiene la construcción de la estructura o carcasa de la válvula motorizada, esta será instalada y acoplada al cuerpo de la válvula, además de montar todos los elementos constitutivos y la realización de las pruebas respectivas, esta tarea tomó 8 días desde el jueves 19 de julio hasta el martes 31 de julio del 2018. En la figura 3.8 -11 se tiene el cronograma completo de las actividades realizadas en la elaboración del proyecto.

Id	Modo de tarea	Nombre de tarea	Duración	Comienzo	Fin	29 abr '18								
						J	V	S	D	L	M	X	J	V
1		Desarrollar una válvula de riego agrícola motorizada programable mediante	66 días	sáb 28/4/18	mar 31/7/18									
2		Investigación, validación de elementos constitutivos de la electroválvula	7 días	sáb 28/4/18	mié 9/5/18									
3		Investigación de elementos electrónicos a utilizar, microcontroladores	2 días	sáb 28/4/18	lun 30/4/18									Juan Mende
4		Investigación de elementos mecánicos a utilizar, servomotor, engranes	2 días	mar 1/5/18	mié 2/5/18									Juan
5		Investigación de módulos bluetooth a utilizar compatibles con Arduino	1 día	jue 3/5/18	jue 3/5/18									Ju
6		investigar programas para desarrollo de aplicación Android	2 días	lun 7/5/18	mar 8/5/18									
7		fin de la tarea	0 horas	mié 9/5/18	mié 9/5/18									
8		Diseñar el sistema mecánico, electrónico, y de comunicaciones de la electroválvula.	8 días	dom 6/5/18	jue 17/5/18									
9		Diseñar el sistema mecánico de la electroválvula	3 días	dom 6/5/18	mar 8/5/18									
10		Diseño del sistema Electrónico	3 días	mié 9/5/18	vie 11/5/18									
11		Diseño del sistema de comunicaciones	3 días	sáb 12/5/18	mar 15/5/18									
12		diseño del acople de los tres sistemas anteriormente mencionados	3 días	lun 14/5/18	mié 16/5/18									

Proyecto: Cronograma de ejecu Fecha: lun 27/8/18	Tarea		Resumen inactivo		Tareas externas	
	División		Tarea manual		Hito externo	
	Hito		solo duración		Fecha límite	
	Resumen		Informe de resumen manual		Progreso	
	Resumen del proyecto		Resumen manual		Progreso manual	
	Tarea inactiva		solo el comienzo			
Hito inactivo		solo fin				

Página 1

Tabla 3.8 Cronograma de Actividades.

Fuente: Elaborado por el Autor

Id	Modo de tarea	Nombre de tarea	Duración	Comienzo	Fin	29 abr '18								
						J	V	S	D	L	M	X	J	V
13		Fin de la tarea	0 horas	jue 17/5/18	jue 17/5/18									
14		Realizar la programación del microcontrolador Arduino nano	22 días	vie 18/5/18	sáb 16/6/18									
15		programar los timer de reloj en Arduino	2 días	vie 18/5/18	dom 20/5/18									
16		programación variables de tiempo entrada y salida	3 días	lun 21/5/18	mié 23/5/18									
17		programación sentencias de apertura de válvula	3 días	jue 24/5/18	dom 27/5/18									
18		programación sentencia de cerrar válvula	3 días	lun 28/5/18	mié 30/5/18									
19		programación calibración de válvula	3 días	jue 31/5/18	dom 3/6/18									
20		programación apertura, cerrar en forma manual	3 días	vie 18/5/18	mar 22/5/18									
21		programación envío de programación a aplicación información	4 días	jue 7/6/18	mar 12/6/18									
22		pruebas de funcionamiento	2 días	mié 13/6/18	jue 14/6/18									
23		Fin de la tarea	0 horas	vie 15/6/18	vie 15/6/18									
24		Crear una aplicación que trabaje con Smartphone sobre sistema operativo Android,	10 días	sáb 16/6/18	vie 29/6/18									
25		Configuración conexión bluetooth	1 día	sáb 16/6/18	sáb 16/6/18									

Proyecto: Cronograma de ejecu Fecha: lun 27/8/18	Tarea		Resumen inactivo		Tareas externas	
	División		Tarea manual		Hito externo	
	Hito		solo duración		Fecha límite	
	Resumen		Informe de resumen manual		Progreso	
	Resumen del proyecto		Resumen manual		Progreso manual	
	Tarea inactiva		solo el comienzo			
	Hito inactivo		solo fin			

Página 2

Tabla 3.9 Cronograma de Actividades.

Fuente: Elaborado por el Autor






Id	Modo de tarea	Nombre de tarea	Duración	Comienzo	Fin	29 abr '18								
						J	V	S	D	L	M	X	J	V
26			configuración parámetros de tiempo	2 días	dom 17/6/18	lun 18/6/18								
27			Configuración para envíos de tiempo de apertura y cerrar hacia Arduino	2 días	mar 19/6/18	mié 20/6/18								
28			configuración parámetros de calibración de válvula	3 días	jue 21/6/18	sáb 23/6/18								
29			Configuración parámetros de apertura, cerrar en forma manual	2 días	dom 24/6/18	lun 25/6/18								
30			programación para recepción de parámetros de configuración	2 días	mar 26/6/18	mié 27/6/18								
31			pruebas de integración smartphone - Arduino	2 días	jue 28/6/18	vie 29/6/18								
32			Construir el sistema mecánico el cual será acoplado y adaptado a las características de la construcción del sistema de engranes	13 días	lun 2/7/18	mié 18/7/18								
33			construcción del sistema de engranes	4 días	dom 1/7/18	mié 4/7/18								
34			construcción del acople de la válvula con el sistema de engranes	3 días	jue 5/7/18	dom 8/7/18								
35			Acoplar sistema de engranes con acople de válvula	6 días	lun 9/7/18	sáb 14/7/18								
36			pruebas de acoples y engranes en funcionamiento	3 días	dom 15/7/18	mar 17/7/18								
37			fin de tarea	0 horas	mié 18/7/18	mié 18/7/18								
38			Construir la estructura o armazón que será instalada sobre la válvula	8 días	jue 19/7/18	mar 31/7/18								
39			construcción de armazón	3 días	jue 19/7/18	dom 22/7/18								

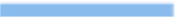


















Proyecto: Cronograma de ejecu Fecha: lun 27/8/18	Tarea		Resumen inactivo		Tareas externas	
	División		Tarea manual		Hito externo	
	Hito		solo duración		Fecha límite	
	Resumen		Informe de resumen manual		Progreso	
	Resumen del proyecto		Resumen manual		Progreso manual	
	Tarea inactiva		solo el comienzo			
Hito inactivo		solo fin				

Página 3

Tabla 3.10 Cronograma de Actividades.

Fuente: Elaborado por el Autor

Id	Modo de tarea	Nombre de tarea	Duración	Comienzo	Fin	29 abr '18								
40		Montaje de armazón sobre la válvula	2 días	lun 23/7/18	mar 24/7/18	J	V	S	D	L	M	X	J	V
41		Instalación de componentes sobre armazón	2 días	mié 25/7/18	jue 26/7/18									
42		Verificación de funcionamiento del sistema completo	2 días	vie 27/7/18	sáb 28/7/18									
43		pruebas de funcionamiento	2 días	dom 29/7/18	lun 30/7/18									
44		fin de tarea	0 horas	mar 31/7/18	mar 31/7/18									

Proyecto: Cronograma de ejecu Fecha: lun 27/8/18	Tarea		Resumen inactivo		Tareas externas	
	División		Tarea manual		Hito externo	
	Hito		solo duración		Fecha límite	
	Resumen		Informe de resumen manual		Progreso	
	Resumen del proyecto		Resumen manual		Progreso manual	
	Tarea inactiva		solo el comienzo			
Hito inactivo		solo fin				

Página 4

Tabla 3.11 Cronograma de Actividades.

Fuente: Elaborado por el Autor

3.14 Ventajas del producto.

El uso de una válvula metálica y de apertura por compuerta, hace que los cambios de presión de flujo no sean tan abruptos, como los sistemas tradicionales que utilizan electroválvulas de solenoide, este sistema está diseñado para grandes presiones.

Se realiza la configuración del dispositivo de forma remota por *Bluetooth* a una distancia promedio de 10 metros, este no necesita ser conectado a un mando central para su configuración como la mayoría de los sistemas actuales de riego.

El dispositivo puede ser configurado a distancia por cualquier *Smartphone* de cualquier marca con sistema operativo Android, con comunicación *Bluetooth*.

Se dispone del módulo de control, comunicación y actuadores en el mismo dispositivo, sin la necesidad de conexión por cable a un mando principal, lo que hace que sea móvil y trabaje de forma autónoma sin la necesidad de depender de otro dispositivo externo.

Se puede disponer de varias de estas válvulas conectados en cascada, con la ayuda de un solo dispositivo *Smartphone* se puede configurar y tener un registro de los parámetros configurados en cada una de estas.

CAPÍTULO IV: IMPLEMENTACIÓN.

En el desarrollo de la electroválvula, fue necesario la adquisición de todos los materiales anteriormente indicados, estos se adquirieron en el mercado nacional, sin necesidad de su importación, la implementación del proyecto se divide en la parte mecánica, parte de control y la parte de programación de aplicación HMI.

Parte mecánica. - Esta parte estará constituida por el servomotor MG-995 y el tren de engranes de una relación de 1:2, este reducirá la velocidad del servo a la mitad y aumentará el torque cercano al doble.

Parte de control. - Esta estará constituida por la electrónica que comprende el microcontrolador Arduino Nano, está es la parte medular de válvula, además de controlar todos los elementos constitutivos de la misma, los elementos electrónicos que se dispone son: módulo *Bluetooth* que realiza la comunicación de la válvula con la plataforma del *Smartphone*, también se dispondrá de un sensor infrarrojo de velocidad, este monitorea el movimiento del vástago de la válvula, el sensor esta como un sensor de fin de carrera.

Parte de *Software*. - Esta hace referencia a la interface HMI, que será la parte del programa, que servirá para programar la válvula con ayuda de un dispositivo *Android*, el programa será realizado con la herramienta de *Appinventor2*, es de acceso web y acceso libre.

Diagrama de bloques.

En la siguiente figura 4.1, se presenta un diagrama de bloques del esquema de funcionamiento de la válvula motorizada.

Se dispone de un dispositivo *Smartphone* el cual se conectará mediante *Bluetooth* con el módulo HC-05, este trabaja en modo esclavo y a su vez con el microcontrolador Arduino Nano, la comunicación será bidireccional entre el dispositivo *Smartphone* y la válvula, el *Smartphone* enviará todos los parámetros de configuración hacia la válvula, como hora del dispositivo, hora de apertura de la válvula, hora de cierre de la válvula, y la opción de armado de esta, y de apertura y cierre manual.

Los parámetros configurados en la válvula pueden ser enviados al dispositivo *Smartphone*, para luego ser almacenados en una base de datos y visualizar los registros de las válvulas configuradas.

Se dispondrá de un sensor de velocidad infrarrojo con un *Encoder* el cual actúa como sensor de fin de carrera, este sensor monitorea la apertura o cierre total de la válvula.

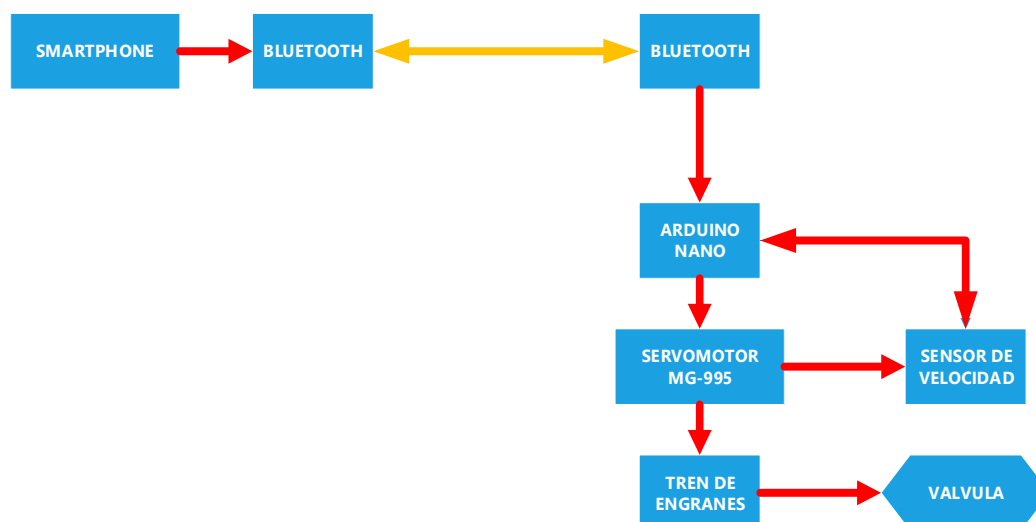


Figura. 4.1. Diagrama de bloques del funcionamiento de la válvula.

Fuente: Elaborado por el Autor

En la figura 4.2, se puede visualizar el diagrama de control de la válvula motorizada, en esta, se puede apreciar los elementos que la componen.

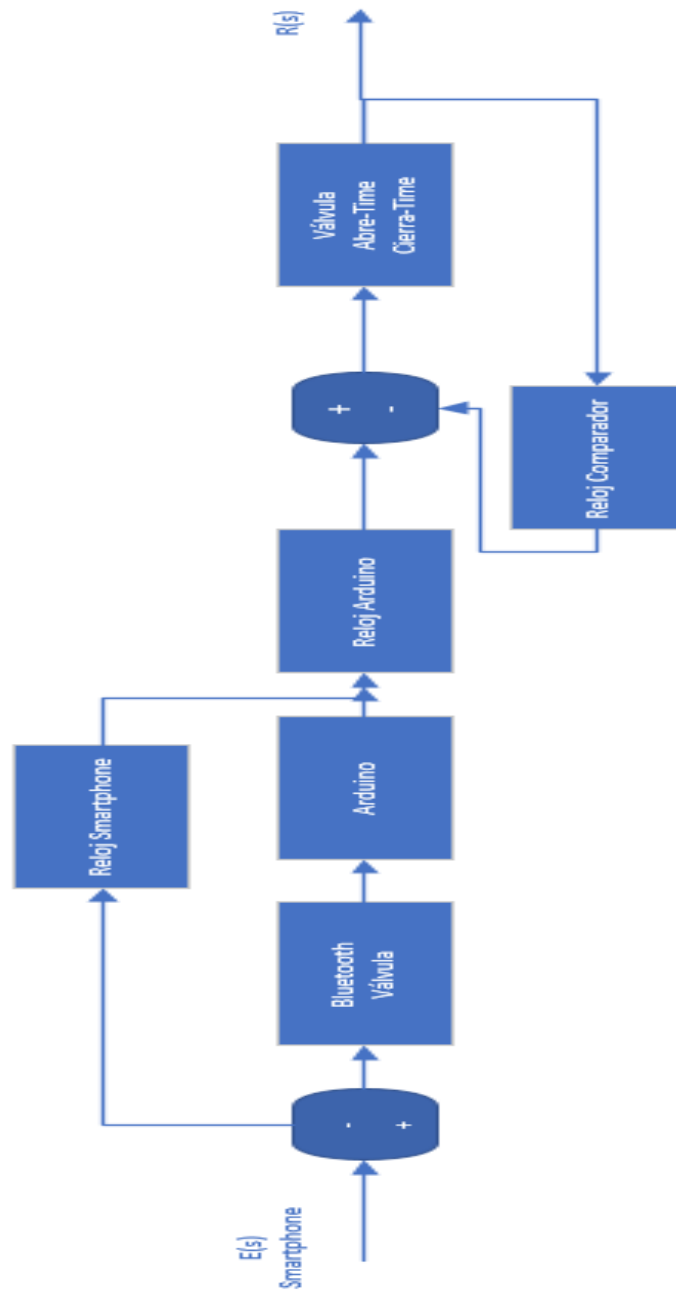


Figura. 4.2 Diagrama de Control.

Fuente: Elaborado por el Autor

4.1 Diseño de la parte mecánica.

En el diseño de la parte mecánica, se construirá el armazón para la instalación de cada una de la parte a ser implementadas, servomotor, sensores, módulos, controlador, sobre

todo el diseño del sistema de engranes, los cuales serán utilizados por el servomotor y el vástago de la válvula.

4.2 Diseño de piezas para el armado de la válvula

Se realiza el diseño e impresión de todas y cada una de las piezas utilizadas en el armado de la válvula, estas piezas fueron impresas en una impresora 3D con materia PLA.

El diseño de la carcasa y soportes, se lo realizó de acuerdo con el modelo y forma física de la válvula, el diseño se acopla exactamente al entorno de la válvula, logrando una sujeción de la carcasa al cuerpo de la válvula de una manera exacta, de igual forma con cada una de las piezas que serán utilizadas en la elaboración de la válvula

Cada una de estas piezas se diseñó en el programa FreeCAD, este permite diseñar piezas mecánicas y no mecánicas, para luego ser exportadas en archivos STL, estos archivos son reconocidos por el software de las impresoras 3D, con ayuda de un impresora 3D y el software adecuado se logró materializar cada de un de estas piezas en materia PLA, como se observa en la figura 4.3.

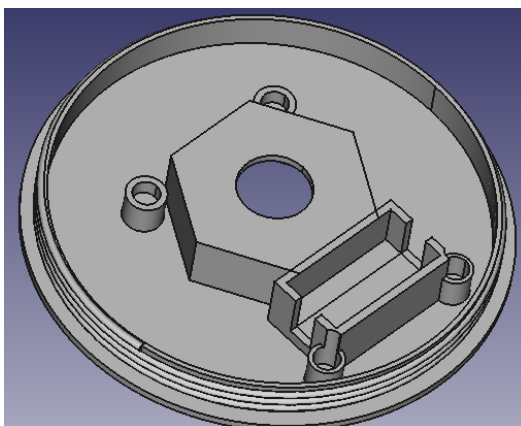


Figura. 4.3. Base de carcasa de la valvular,

Fuente: Elaborado por el autor

4.2.1 Diseño de la base inferior de la carcasa.

En la figura 4.4, se encuentra la válvula que fue modificada para convertirla en una válvula motorizada y programable, en las figuras 4.5, 4.6 y 4.7 se puede verificar el diseño de la base, que fue acoplada a la válvula metálica.



Figura. 4.4. Válvula de 2 pulgadas.

Fuente: Elaborado por el Autor.

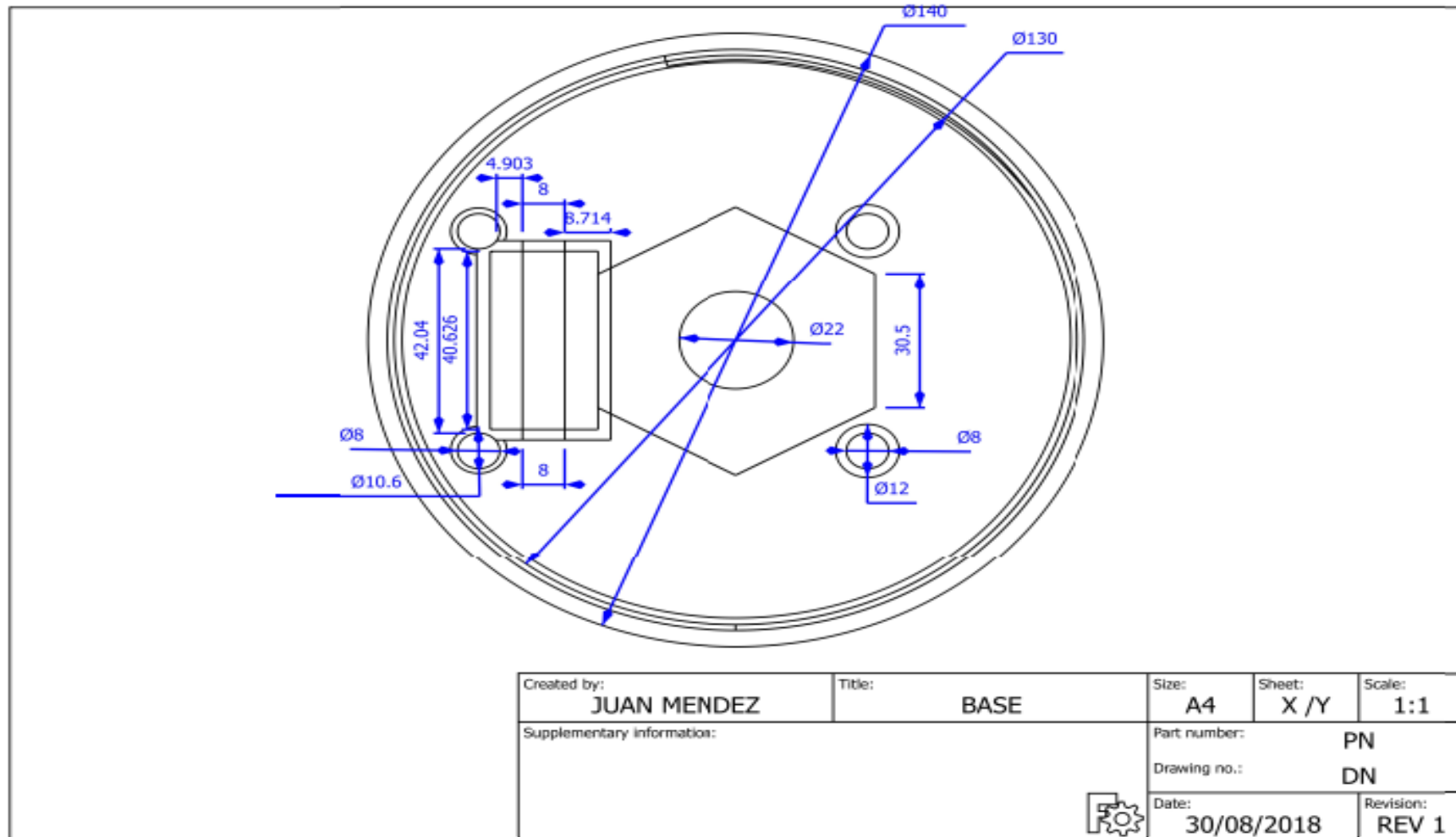
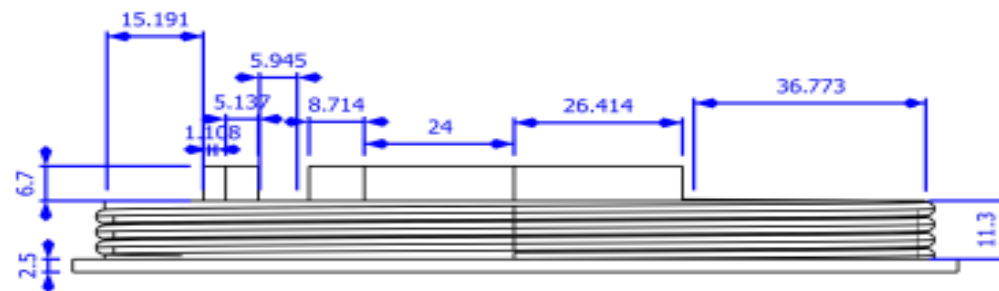


Figura. 4.5 Diseño 3D base vista superior válvula.

Fuente: Elaborado por el Autor.



Created by: JUAN MENDEZ	Title: BASE LATERAL	Size: A4	Sheet: X / Y	Scale: 1:1
Supplementary information:		Part number: PN		
		Drawing no.: DN		
		Date: 30/08/2018	Revision: REV 1	

Figura. 4.6 Diseño 3D base vista lateral válvula.

Fuente: Elaborado por el Autor.

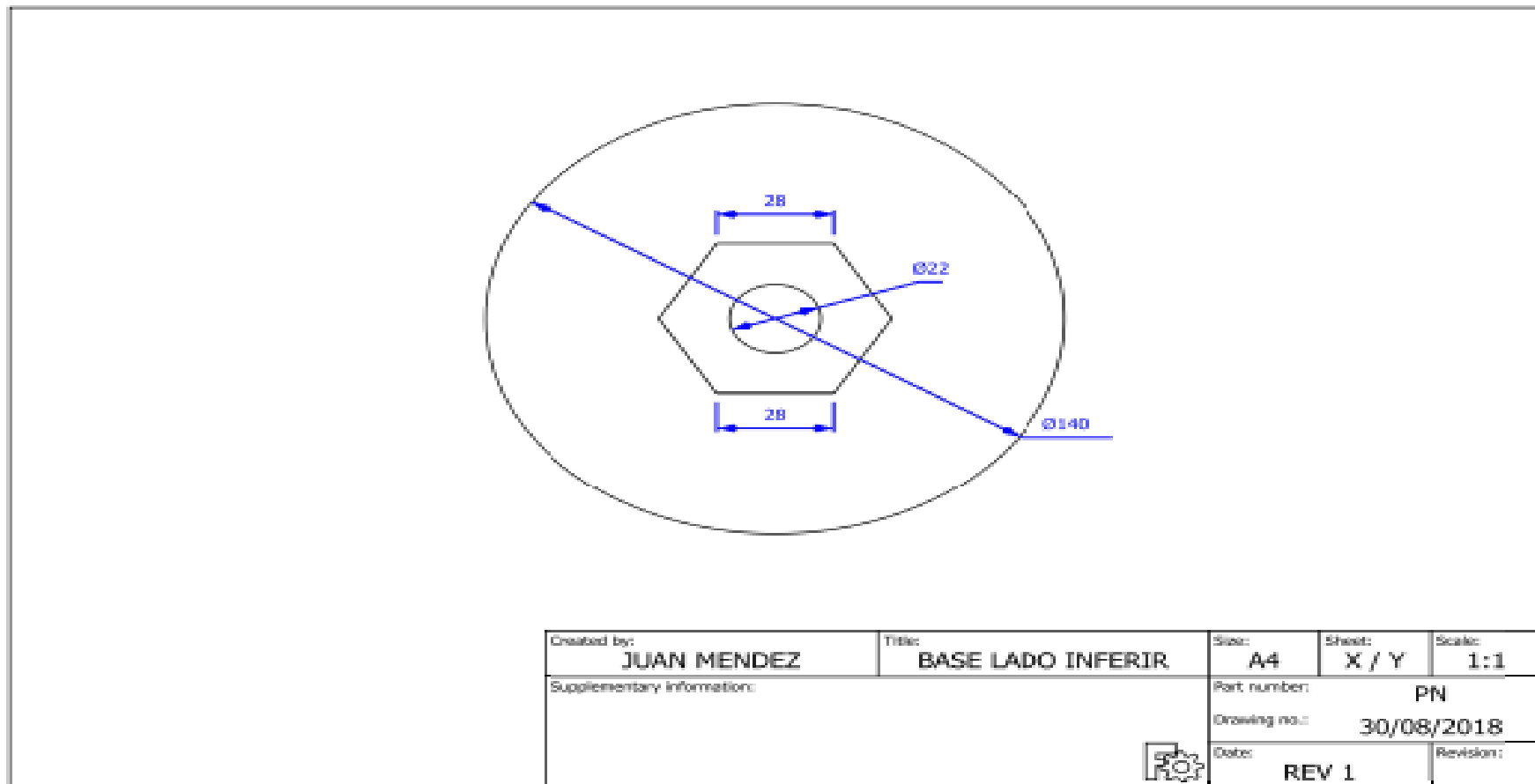


Figura. 4.7 Diseño 3D base vista inferior válvula.

Fuente: Elaborado por el Autor

4.2.2 Diseño del tren de engranes.

Para poder efectuar la apertura y cierre de la válvula, por medio del vástago de la válvula, se realizó el diseño de un tren de engranes, este permitirá el aumento del torque suministrado por el servomotor al doble y una disminución de la velocidad a la mitad.

Para el cálculo de la relación de transferencia del torque y velocidad, se utilizó la relación del número de dientes, se tiene una relación de número de dientes de 2:1, se dispone de un engranes de 10 y 20 dientes en el servomotor y el vástago de la válvula, con esta relación se logra disponer de aproximadamente el doble del torque del servomotor y disminuir la velocidad a la mitad, como se observa en la figura 4.8.

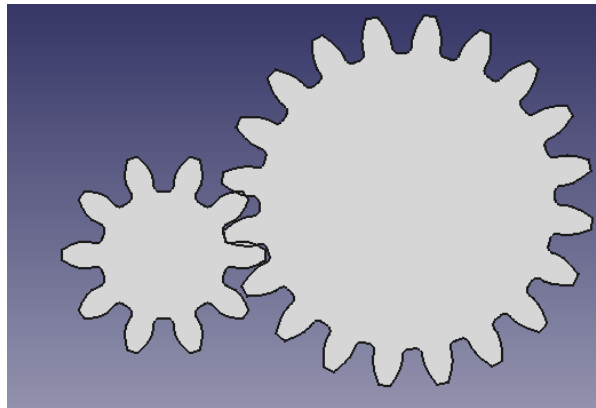


Figura. 4.8 Tren de engranes.

Fuente: Elaborado por el autor

4.2.3 Cálculo del torque para el vástago de la válvula.

Es necesario realizar el cálculo de la fuerza necesaria para mover el vástago de la válvula, este cálculo se lo realiza con la fórmula de fuerza, para mover el disco de la válvula fue necesario suministrar una masa de 100g, ver figura 4.9

Formula del cálculo de la fuerza

$$F = m \cdot g$$

$F = \text{fuerza (N)}$

$m = \text{masa (Kg)}$

$g = \text{aceleración de la gravedad (m/s}^2\text{)}$

$$F = 0.1\text{Kg} \times 9,8 \text{ m/s}^2$$

$$\underline{F = 0,98 \text{ N}}$$

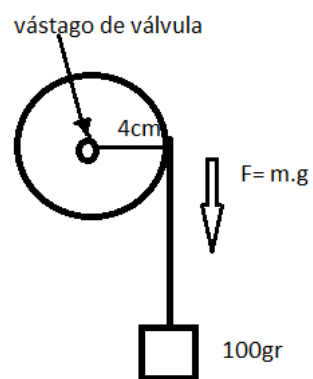


Figura. 4.9 Cálculo torque.

Fuente elaborado por el Autor

Para el cálculo del torque, se lo realiza con la fuerza anteriormente calculada, el radio del disco es de 10cm, y aplicando la fórmula correspondiente del cálculo de torque (cálculo del torque necesario para mover el vástago de la válvula)

$$T = F \cdot r$$

$T = \text{torque (Nm)}$

$F = \text{Fuerza (N)}$

$r = \text{radio del disco (m)}$

$$T = 0,98\text{N} \times 0,04\text{m}$$

$$\underline{T = 0,0392 \text{ Nm.}}$$

$$\underline{T = 3,92 \text{ Kgf-cm.}}$$

4.2.4 Cálculo de relación de transmisión.

Para el cálculo de la relación de transmisión, se lo realiza con la ecuación de relación de transferencia, esta hace referencia al número de dientes de los engranes y a la velocidad de revolución en (RPM), como se observa en la figura 4.10.

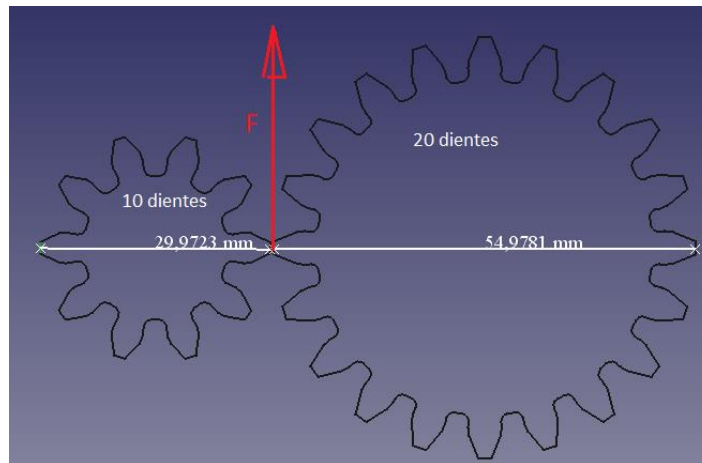


Figura. 4.10 Relación entre engranes.

Fuente: Elaborado por el Autor.

$r_1 = 29.9 \text{ mm}$ radio engrane servo

$r_2 = 54.9 \text{ mm}$ radio engrane vástago

$Z_2 =$ número de dientes engrane vástago

$Z_1 =$ número de dientes engrane servo

$n_2 =$ velocidad angular en rpm del engrane vástago

$n_1 =$ velocidad angular en rpm del engrane servo

$r_e =$ relación de transmisión.

De acuerdo con el *datasheet* del servomotor tiene una velocidad de $0,2 \text{ s}/60^\circ (4,8\text{V}) = 50 \text{ rpm}$.

$$r = Z_2/Z_1 = n_1/n_2$$

$$r = 20/10 = 2$$

$$2 = \frac{n_1}{n_2}$$

$$n_2 = \frac{n_1}{2}$$

$$n_2 = 50 \frac{rpm}{2} = 25 rpm$$

Es necesario disponer del cálculo del torque para mover el vástago de la válvula, este cálculo se lo realiza de la siguiente manera, utilizando la ecuación correspondiente del torque en engranes

Torque del servomotor de acuerdo con el *datasheet*.

$$T_1 = 8,5 \text{ kgf}\cdot\text{cm} (4.8 \text{ V})$$

$$r_1 = 15 \text{ mm}$$

$$r_2 = 27,5 \text{ mm}$$

$$T_1 = 8,5 \text{ kgf}\cdot\text{cm} (4.8 \text{ V})$$

Cálculo del torque transmitido al engrane para mover el vástago de la válvula.

$$T = F \cdot r$$

$$F_1 = F_2$$

$$F = \frac{T_1}{r_1} = \frac{T_2}{r_2}$$

$$T_2 = \left(\frac{T_1}{r_1} \right) r_2$$

$$T_2 = \left(\frac{8,5 \text{ Kg}\cdot\text{cm}}{1,5 \text{ cm}} \right) \times 2,75 \text{ cm}$$

$$T_2 = 15,5 \text{ Kg}\cdot\text{cm}$$

En las figuras 4.11, 4.12 y 4.13 se puede ver el diseño del tren de engranes, además de la base sobre la cual estarán acoplados.

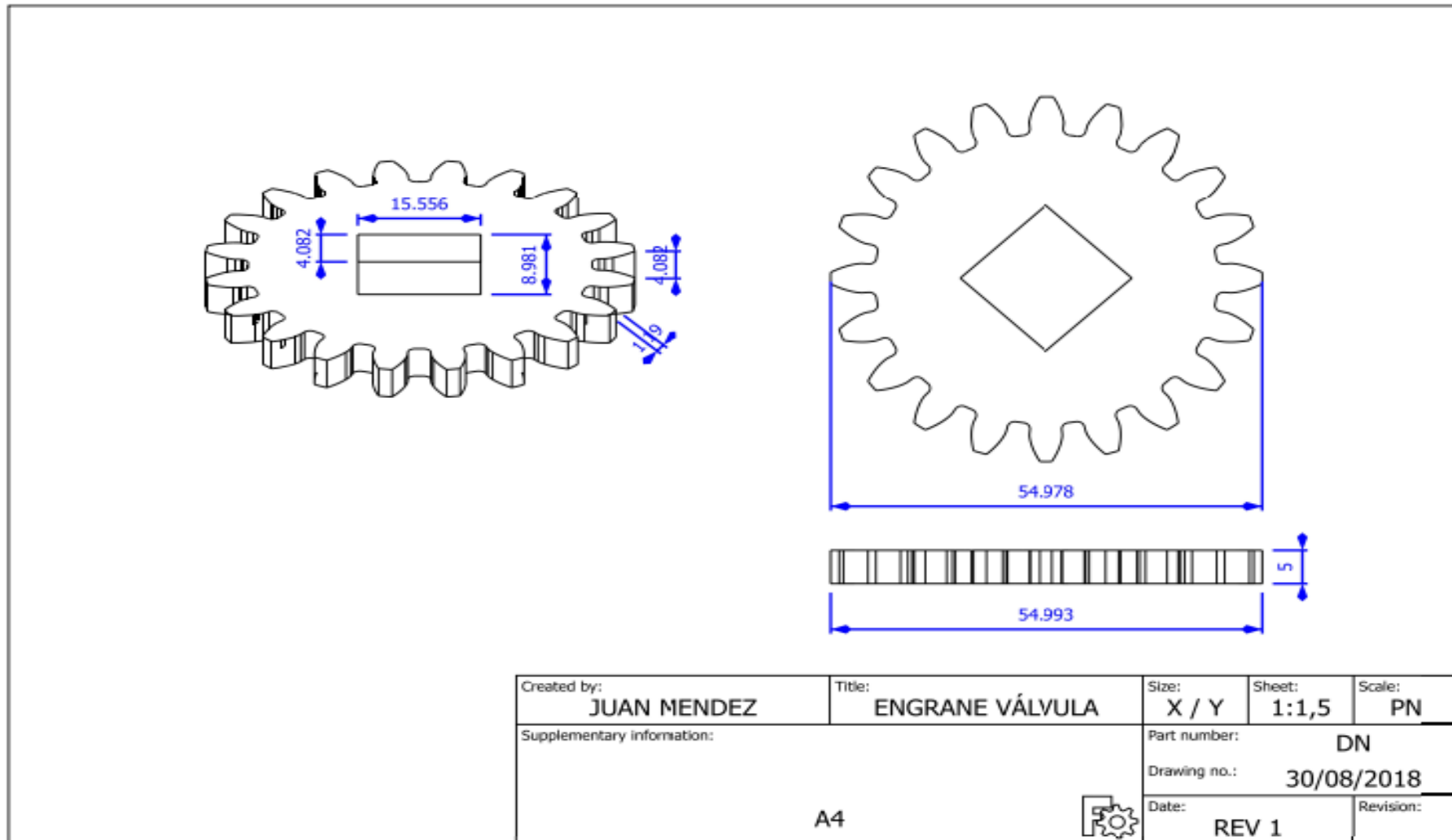


Figura. 4.11 Diseño 3D engrane válvula.

Fuente: Elaborado por el Autor.

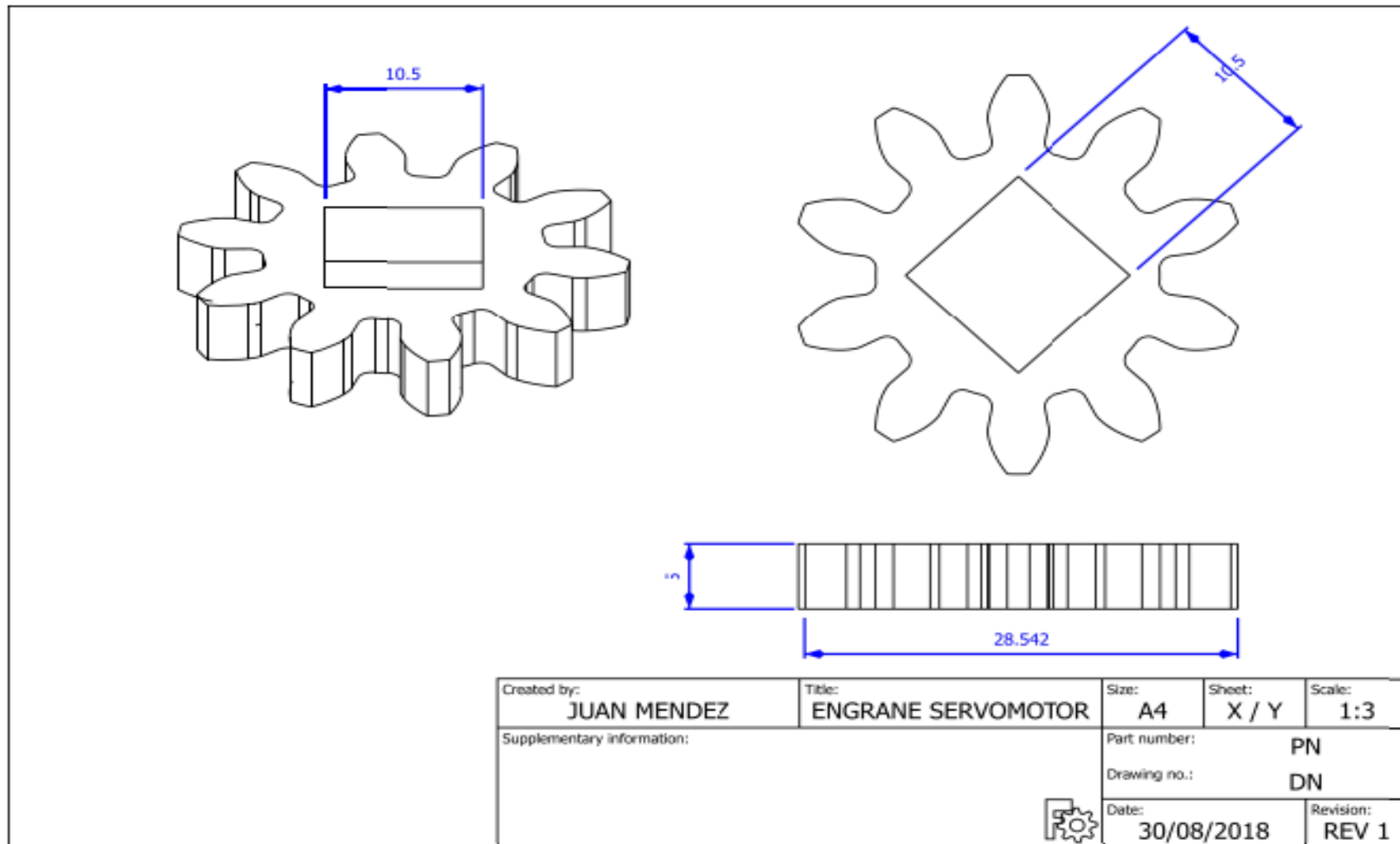


Figura. 4.12 Diseño 3D engrane servomotor.

Fuente: Elaborado por el Autor.

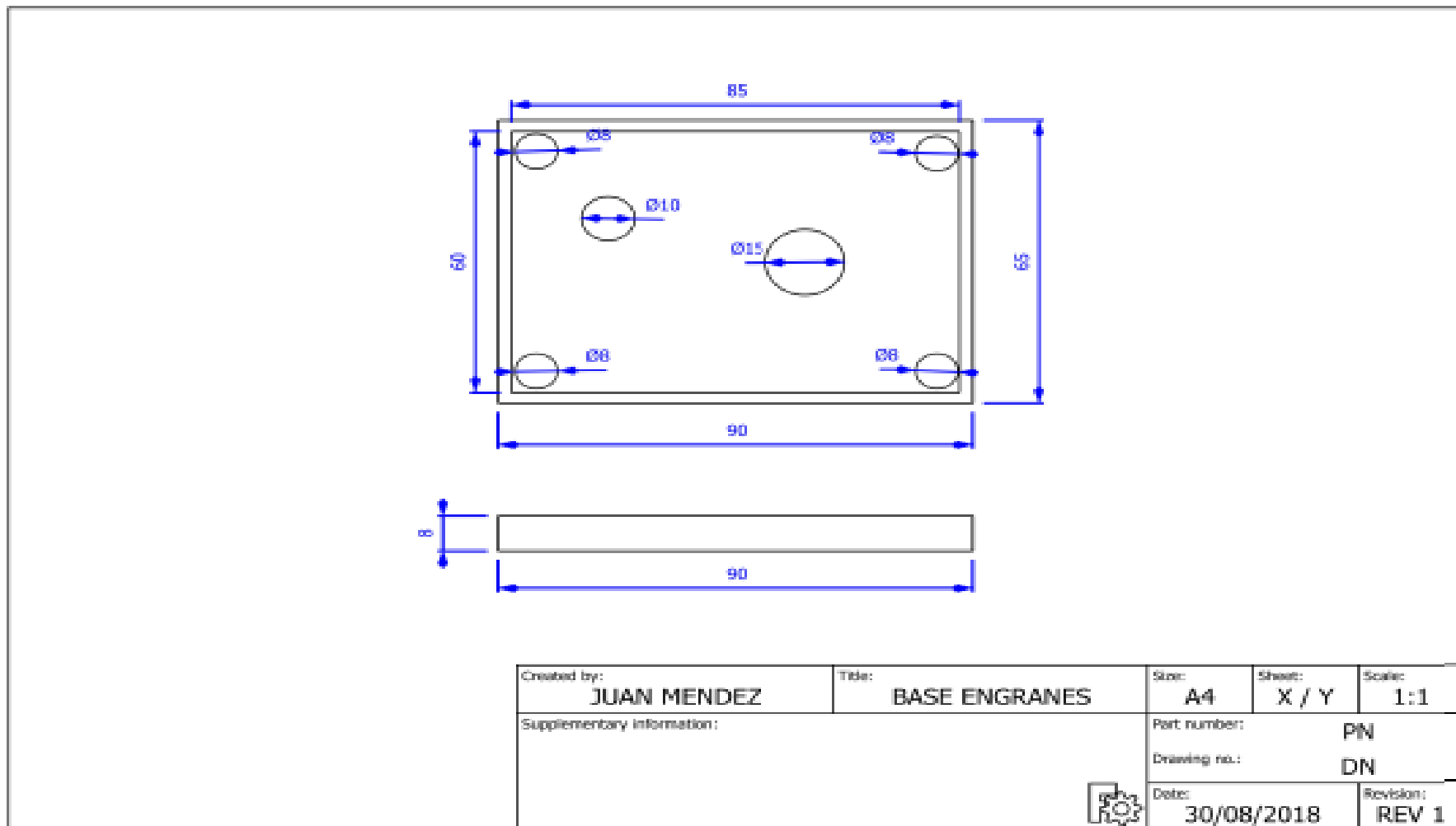


Figura. 4.13 Diseño 3D base engranes.

Fuente: Elaborado por el Autor.

4.3 Diseño del encoder.

En la figura 4.14, se puede ver el diseño del encoder utilizado para el sensor de velocidad, este encoder será instalado sobre el soporte del vástago de la válvula.

4.4 Diseño de la base para las baterías.

En la figura 4.15, se encuentra el diseño de la base de baterías, sobre esta base serán instaladas las 5 celdas que suministrarán la energía a la operación de la válvula, además de cumplir el papel de arreglar, ordenar y dar estética al dispositivo.

4.5 Diseño base baterías.

Para el diseño de la carcasa superior de la válvula, se puede apreciar en la figura 4.16, esta viene a ser la tapa superior en la cual se encuentra instalado el panel solar, esta carcasa dará una protección a todos los componentes internos de la válvula.

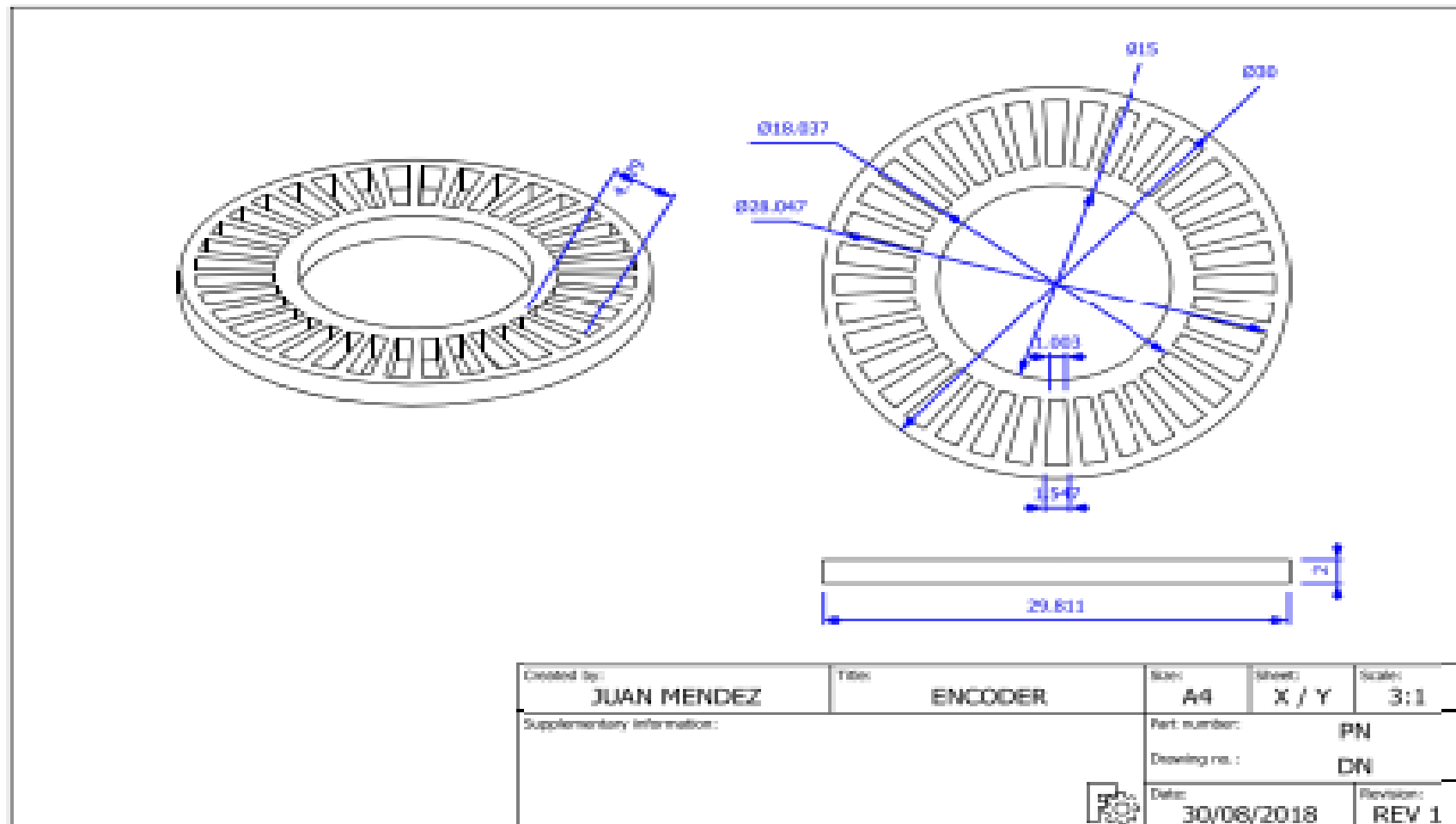


Figura. 4.14 Diseño 3D de encoder.

Fuente: Elaborado por el Autor.

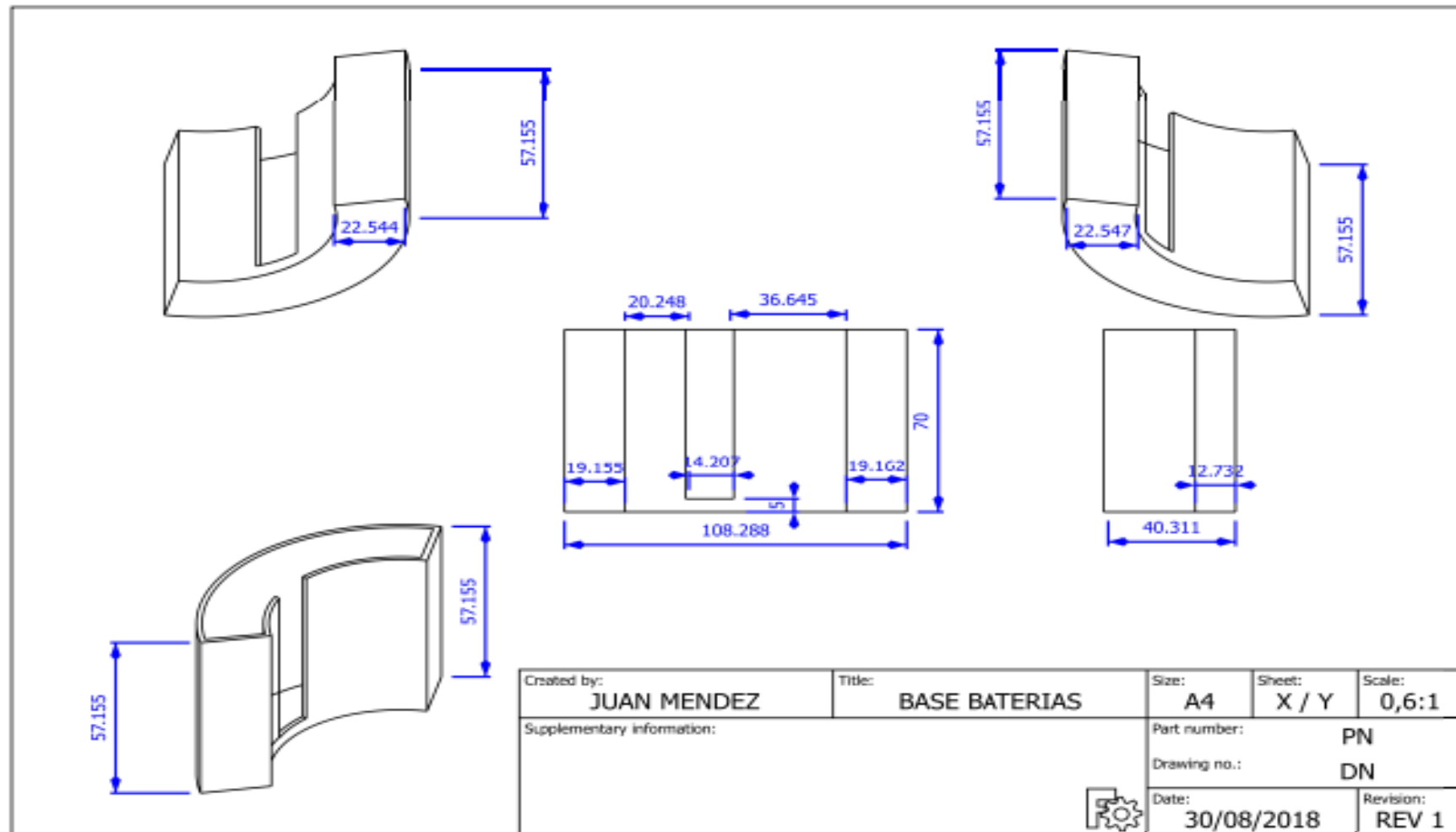


Figura. 4.15 Diseño 3D base baterías.

Fuente: Elaborado por el Autor.

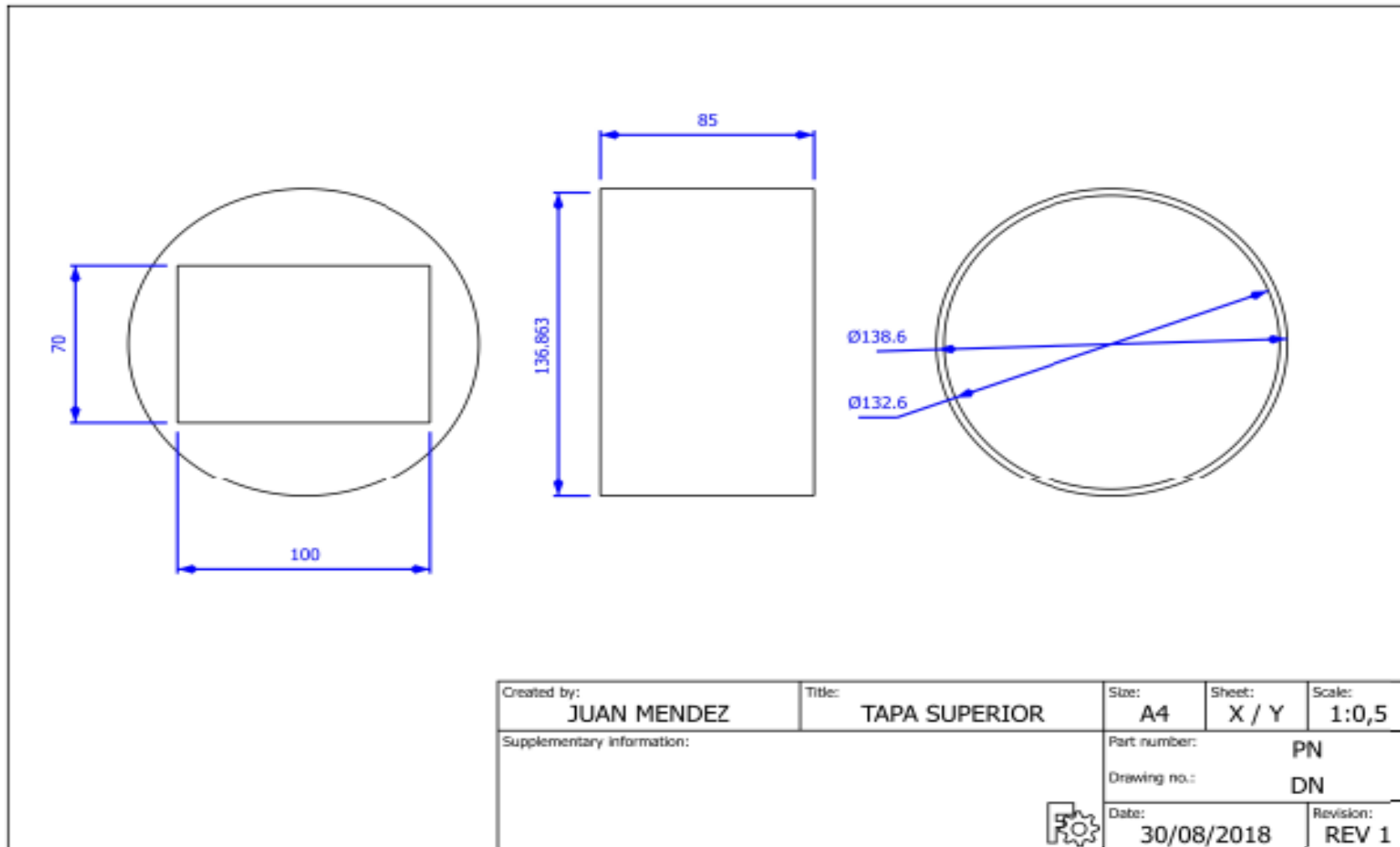


Figura. 4.16 Diseño 3D carcasa superior.

Fuente: Elaborado por el Autor.

4.6 Diseño electrónico.

En la figura 4.17, se muestra el diseño electrónico, en este se tiene el microcontrolador Arduino Nano, el cual es la parte medular de la válvula, en el microcontrolador se encuentra toda la programación para trabajar con el módulo *Bluetooth*, sensor de velocidad, servomotor, y con el aplicativo del dispositivo *Smartphone*.

El microcontrolador Arduino Nano, interactúa de manera directa con cada uno de los sensores, módulos, y servomotor, estos trabajarán con funciones específicas, y el conjunto de todos estos, realizarán el funcionamiento del dispositivo.

4.7 Dimensionamiento de fuente de alimentación del circuito.

Para el correcto funcionamiento del circuito, es necesario seleccionar la fuente de alimentación correcta, para el análisis del consumo de corriente de cada uno de los elementos que conforman el circuito, se lo realiza de forma teórica con los datos obtenidos de los *datasheet* de cada elemento, y de forma práctica con ayuda de un multímetro, además del voltaje necesario para el correcto funcionamiento de sus elementos, como se observa en la figura 4.17.

En la tabla 4.1, se presentan los valores de consumo de corriente teórico y prácticos de los elementos de la válvula motorizada.

	Arduino Nano	Bluetooth	Sensor Infrarrojo	Servomotor	LED	Total
Consumo Teórico	15 mA	50 mA	20 mA	100 mA	15 mA	200 mA
Consumo Medido	23 mA	36 mA	28 mA	110mA	15 mA	212 mA
Voltaje	5V	5V	5V	5V		5V

Tabla 4.1 Consumo de corriente teórica y práctica.

Fuente: Elaborado por el Autor.

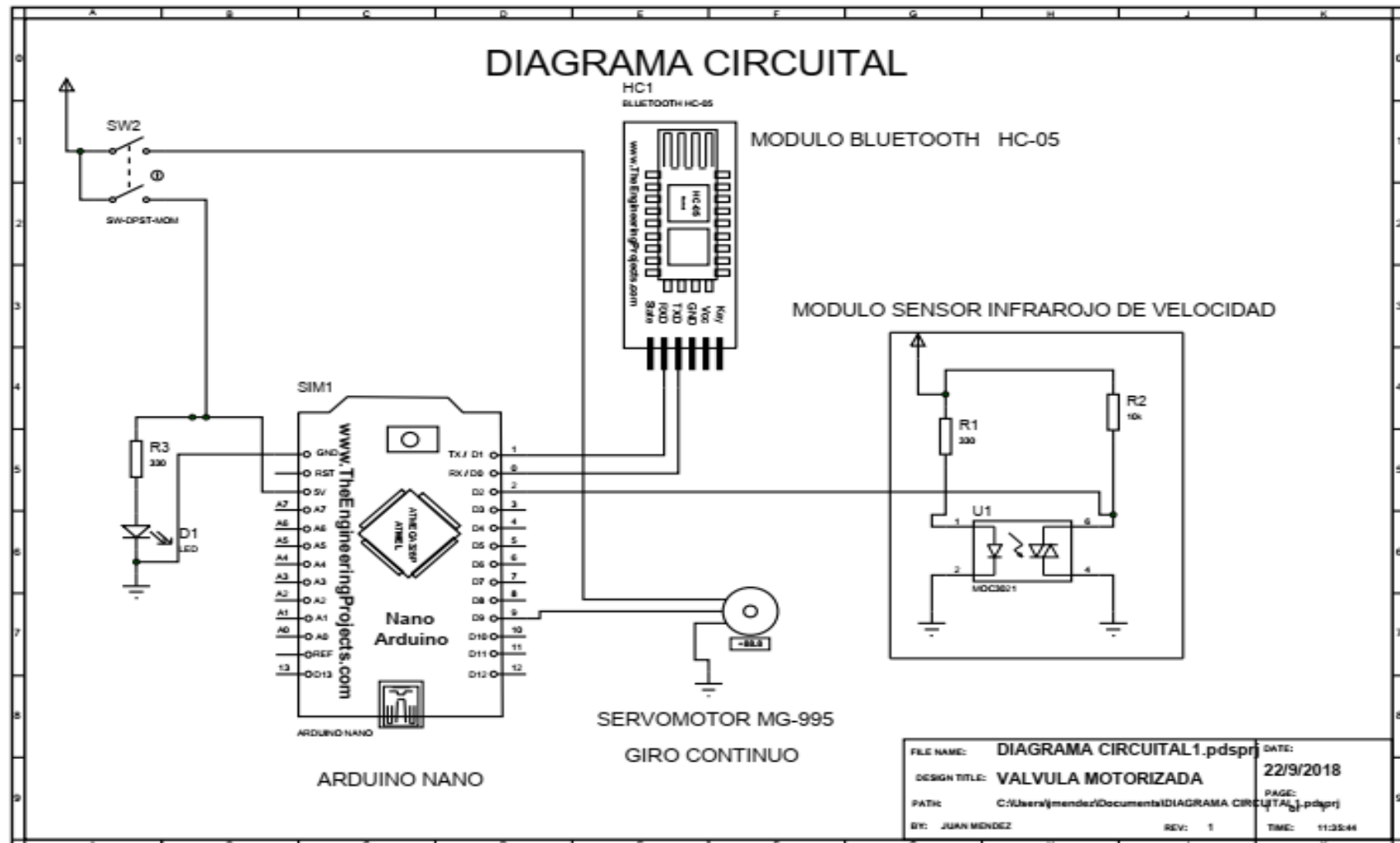


Figura. 4.17 Diagrama circuital.

Fuente: Elaborado por el Autor.

Con los consumos de corriente total del circuito, se busca en el mercado nacional, una batería que pueda solventar la cantidad de corriente que se requiere y se ajuste al presupuesto.

En el mercado nacional se dispone de baterías o *Powerbank* de 5V a 4000mAh de 5 celdas, este tipo de baterías esta de acorde con el consumo de corriente de todo el circuito, ya que este tiene un consumo máximo de 212mA.

$$P = 4000mA \times 5V = 20 W$$

Cálculo del tiempo promedio de funcionamiento de la válvula.

$$\text{Tiempo de funcionamiento} = \frac{4000mAh}{212 \text{ mA}} = 18,8 \text{ Horas}$$

El valor obtenido es de alrededor de 18,8 horas es el tiempo máximo de funcionamiento que tiene todo el circuito con la batería que se va a utilizar.

Se dispone de un LED indicador de encendido de la válvula, se realiza los cálculos de la resistencia de protección del LED, el consumo de corriente para de un LED de color rojo es de 15mA de acuerdo con el *datasheet* de este tipo.

$$R = \frac{V}{I}$$

$$R = \frac{5V}{15mA} = 333,3 \Omega$$

En el mercado se dispone de resistencias de 330 Ω , la cual es utilizada para protección del LED.

Para el cálculo del tiempo utilizado en la carga de una batería con un adaptador de corriente de 1000 mA, se los realiza utilizando la formula correspondiente.

$$\text{Tiempo de carga de batería} = \frac{4000mAh}{1000mA} = 4 \text{ horas}$$

4.7.1 Arduino Nano y sensor de velocidad HC-03.

El microcontrolador Arduino Nano con el sensor de velocidad HC-03, trabajarán de forma conjunta para censar el giro del vástago de la válvula, es decir el sensor infrarrojo censará el movimiento del vástago con ayuda de un *encoder*, mientras se esté censando el movimiento en el vástago, el Arduino Nano sigue con la señal *PWM* hacia el servomotor el cual girará en la posición indicada.

Si el vástago de la válvula deja de girar al alcanzar la apertura o cierre total de la válvula, el sensor de velocidad dejará de censar el movimiento del *encoder* y en ese instante el módulo Arduino no aplicará una señal *PWM* al servomotor, esto hará que el servomotor deje de girar, como se puede observar en la figura 4.18.

El sensor infrarrojo dispone de cuatro pines, dos de los cuales son de alimentación, pin 1 VCC, pin 2 GND y el pin 3 de una salida digital, este pin digital es conectado al pin 2 del Arduino, como se observa en la tabla 4.2

PIN Sensor Infrarrojo	PIN Arduino
1 VCC	
2 GND	
3 D0	D2

Tabla 4.2 Tabla conexión infrarrojo y Arduino Nano.

Fuente: Elaborado por el Autor.

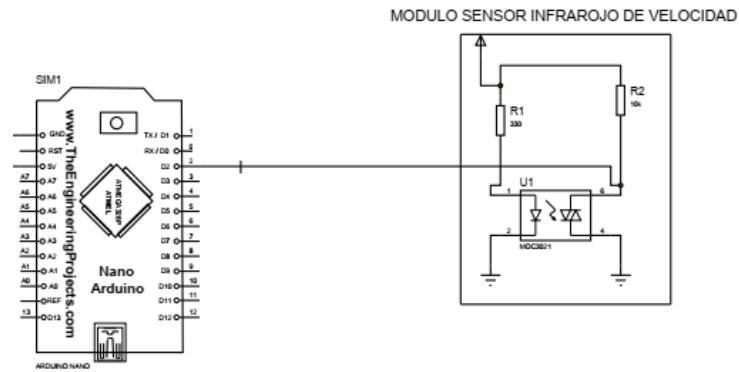


Figura. 4.18 Diseño sensor velocidad HC-03 y Arduino.

Fuente: Elaborado por el Autor.

4.7.2 Arduino Nano y Módulo *Bluetooth* HC-05.

El conjunto del microcontrolador Arduino Nano y el módulo *Bluetooth* realizan la parte de comunicación entre el dispositivo *Smartphone* y la válvula, el módulo *Bluetooth* trabaja en modo *Slave* con lo cual se garantiza que cualquier dispositivo *Smartphone* pueda conectarse con el módulo *Bluetooth*, siempre que se disponga de las credenciales indicadas (*Password*), como se observa en la figura 4.19.

En la tabla 4.3, se puede validar la conexión de los pines del módulo *Bluetooth* con el microcontrolador Arduino Nano, el pín Tx del módulo HC-05 con el pín Rx de Arduino, el pín RX de módulo HC-05 con el pín Tx del Arduino Nano, también dispone de dos pines de alimentación VCC y GND.

PIN Módulo HC-05	PIN Arduino
2 VCC	
3 GND	
4 TX	RX
5 RX	TX

Tabla 4.3 Tabla conexión HC-05 y Arduino Nano.

Fuente: Elaborado por el Autor.

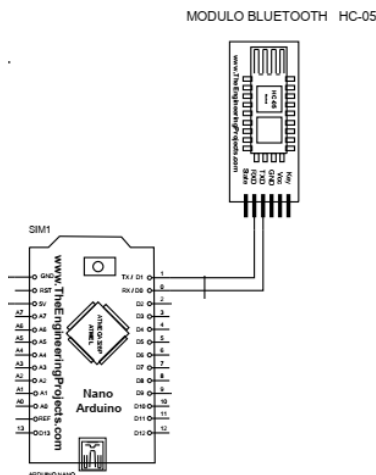


Figura. 4.19. Diseño módulo *Bluetooth* y Arduino,

Fuente: Elaborado por el autor

4.7.3 Arduino Nano y servomotor MG-995.

La combinación del microcontrolador Arduino Nano y el servomotor MG-995, realizara la función de abrir y cerrar la válvula a la hora deseada, el servomotor MG-995 está conectado directamente a la batería, este tiene un consumo considerable y el controlador Arduino no puede suministrarla, únicamente aplicará la señal PWM al servomotor, por ser una corriente que sí puede suministrar el controlador Arduino Nano, como se observa en la figura 4.20.

En la tabla 4.4, se puede validar la conexión de los pines del servomotor MG-995 y el microcontrolador Arduino Nano, los pines de alimentación del servomotor son alimentados directamente desde la batería, y el pin de señal PWM es conectado al pin D9 del Arduino.

PIN Servo MG-995	PIN Arduino
1 VCC	
2 GND	
3 PWM	D 9

Tabla 4.4 Tabla conexión HC-05 y Arduino Nano.

Fuente: Elaborado por el Autor.

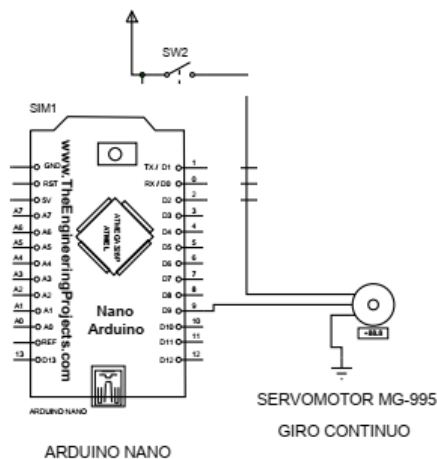


Figura. 4.20. Comunicaciones Arduino Nano- servomotor

Fuente: Elaborado por el Autor

4.8 Diseño de la parte de software.

El desarrollo de la aplicación está enfocada a configurar todos los parámetros de la válvula motorizada, para ello se requiere de un diagrama de flujos en el cual se especifique los pasos de comunicación de la App.

La aplicación se inicializa con el encendido de esta, y la activación de la comunicación *Bluetooth* entre la App y el controlador Arduino, estos se comunican mediante una conexión punto a punto, para lo cual el *Smartphone* se engancha o empareja con el módulo *Bluetooth* de la válvula.

Una vez realizado la comunicación del dispositivo *Smartphone* con el módulo Bluetooth de la válvula, se prosigue con la configuración de todos los parámetros de la válvula.

4.8.1 Diseño de aplicación conexión Bluetooth.

Para una mejor comprensión de la conexión *Bluetooth*, se realiza el siguiente diagrama de flujo, este inicia con el encendido de la aplicación, a continuación de selecciona la válvula

en la lista de dispositivo *Bluetooth*, una vez seleccionado la válvula se procede con la conexión de esta, como se aprecia en la figura 4.21.

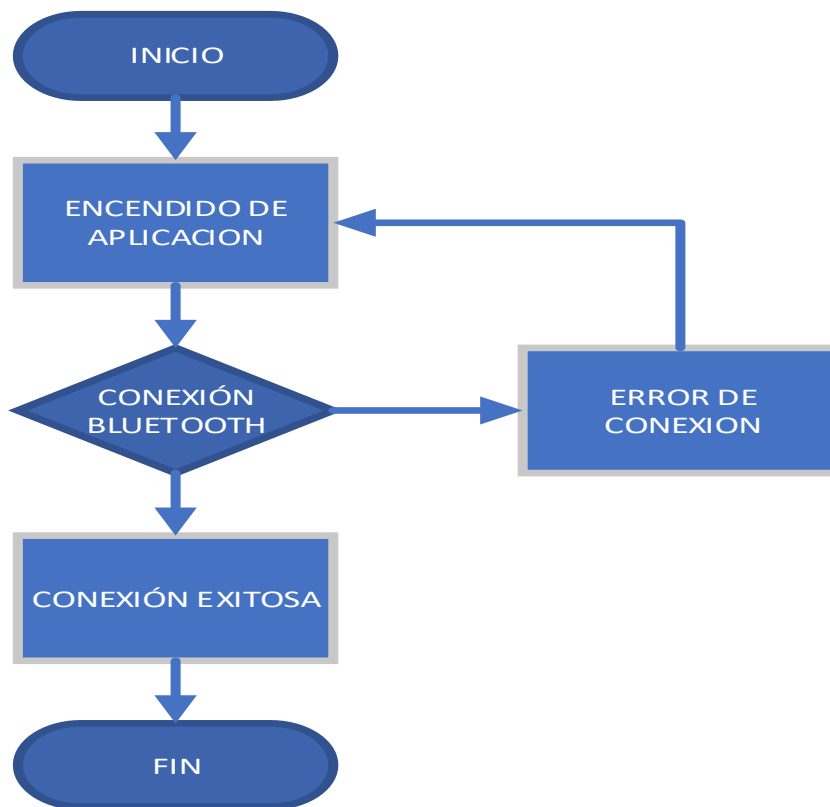


Figura. 4.21 Diagrama de flujo de conexión Bluetooth.

Fuente: Elaborado por el Autor.

4.8.2 Conexión *Bluetooth*.

En la conexión *Bluetooth*, se dispone en el menú de un botón que despliega los dispositivos *Bluetooth* que se encuentran configurados en el dispositivo *Smartphone*, también de un botón de conexión del dispositivo *Bluetooth* seleccionado, y de un botón de desconexión del dispositivo *Bluetooth*, como se observa en la figura 4.22.

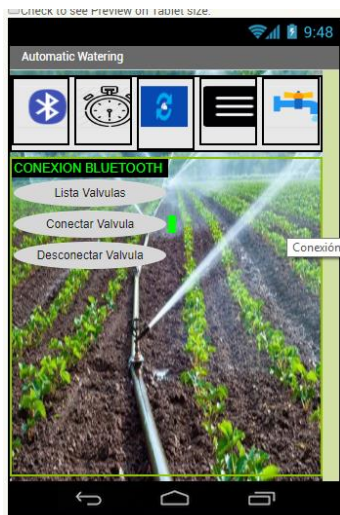


Figura. 4.22. Conexión *Bluetooth* con válvula,

Fuente: Elaborado por el Autor.

En la configuración del menú de conexión *Bluetooth* se dispone de tres botones, con las siguientes opciones.

Lista Válvulas. - Con este botón se despliega la lista de dispositivos *Bluetooth* que se encuentran al alcance del Smartphone, y se selecciona la válvula a la que se desea configurar, además se despliega la información de la válvula seleccionada en el casillero correspondiente.

Conectar Válvula. - Con este botón se conecta la válvula seleccionada, además se despliega el nombre de la válvula seleccionada y se realiza la conexión, de igual forma indica un mensaje de conexión satisfactoria.

Desconectar Válvula. - Con esta opción se desconecta la válvula anteriormente conectada, luego de la desconexión se puede conectar a otra válvula, para proceder con la configuración de esta nueva válvula, como se observa en la figura 4.35.

En la programación del menú del botón de conexión *Bluetooth*, se habilita únicamente lo concerniente a dicho menú, mientras que los otros menús son deshabilitados.

4.8.3 Diseño del menú de configuración horas.

Se dispone de la opción de configurar la hora del *Smartphone* hacia la válvula, con esta opción se logra igualar el reloj interno de la válvula con la hora actual del *Smartphone*, así también se tiene las opciones de la configuración de la hora de apertura y cierre de la válvula, además de la opción de armar la válvula es decir aplicar los valores configurados y aplicarlos en la configuración de esta, como se observa en la figura 4.23.



Figura. 4.23. Opción de igualar reloj.

Fuente: Elaborado por el Autor.

En el siguiente diagrama de flujo, se detalla el funcionamiento de la programación de las horas a configurar, este inicia con la conexión *Bluetooth*, a continuación, se procede a igualar la hora actual, hora de apertura y cierre, además de la opción de armar la válvula, con todos estos datos ingresados el dispositivo está configurado, como se puede apreciar en la figura 4.24.

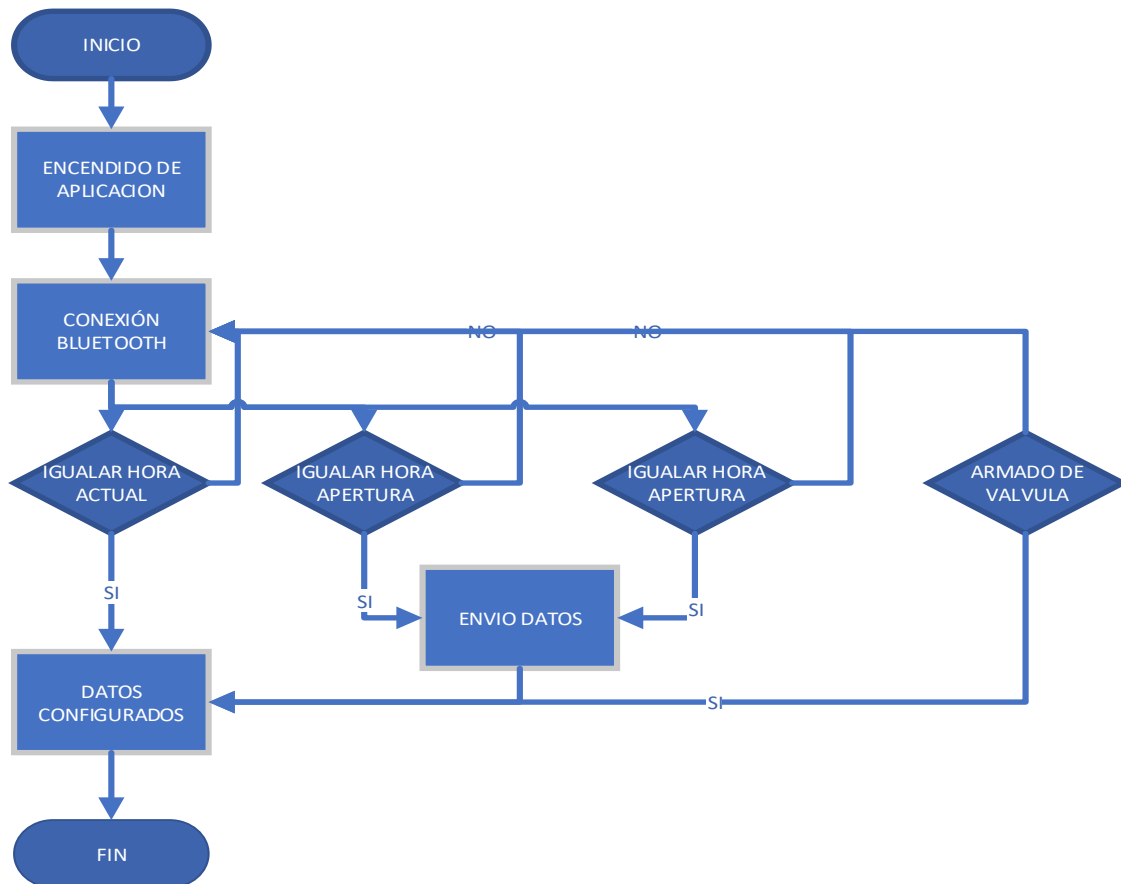


Figura. 4.24 Diagrama de flujo de igualar hora,

Fuente: Elaborado por el Autor

En la programación del menú del botón de Configurar Válvula se habilita únicamente lo concerniente a dicho menú, mientras que los otros menús son deshabilitados momentáneamente.

4.8.4 Diseño de visualización de parámetros configurados.

Se dispone de la opción de visualizar los parámetros configurados en la válvula, como la hora de reloj, apertura, cierre y armado, con esta opción se puede validar que la válvula se encuentre configurada o no, además de registrar y guardar en una base datos, cómo se muestra en la figura 4.25.

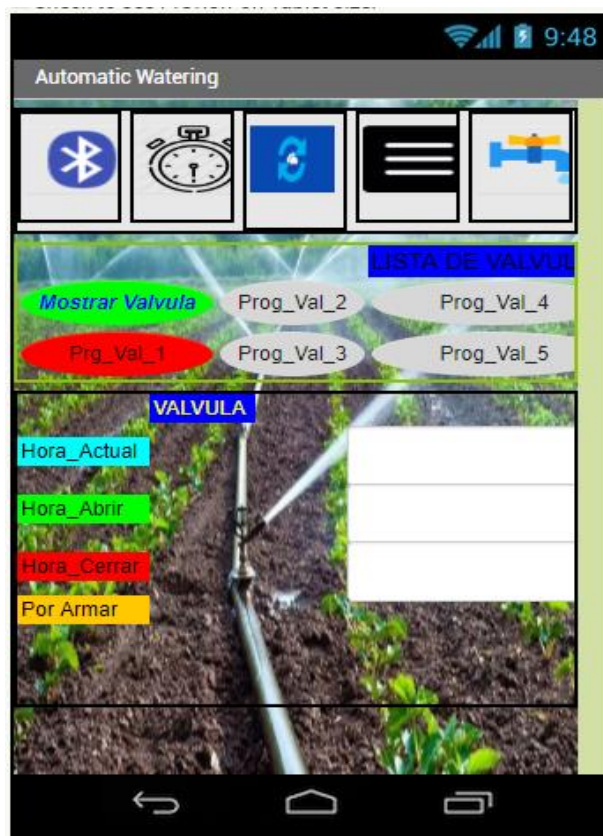


Figura. 4.25. Menú visualización datos para grabar,

Fuente: Elaborado por el Autor

En el siguiente diagrama de flujo, se detalla el funcionamiento de la visualización de los datos ingresados en la válvula, este comienza con la conexión Bluetooth, a continuación, se tiene la opción de mostrar datos, en esta opción se despliegan los datos configurados como hora actual, apertura y cierre, estos pueden ser grabados en los cinco registros, para luego ser visualizados en una tabla.

En la figura 4.26 se muestra en detalle el diagrama de flujo para la visualización de los parámetros configurados en la válvula y luego ser grabados en una base de datos.

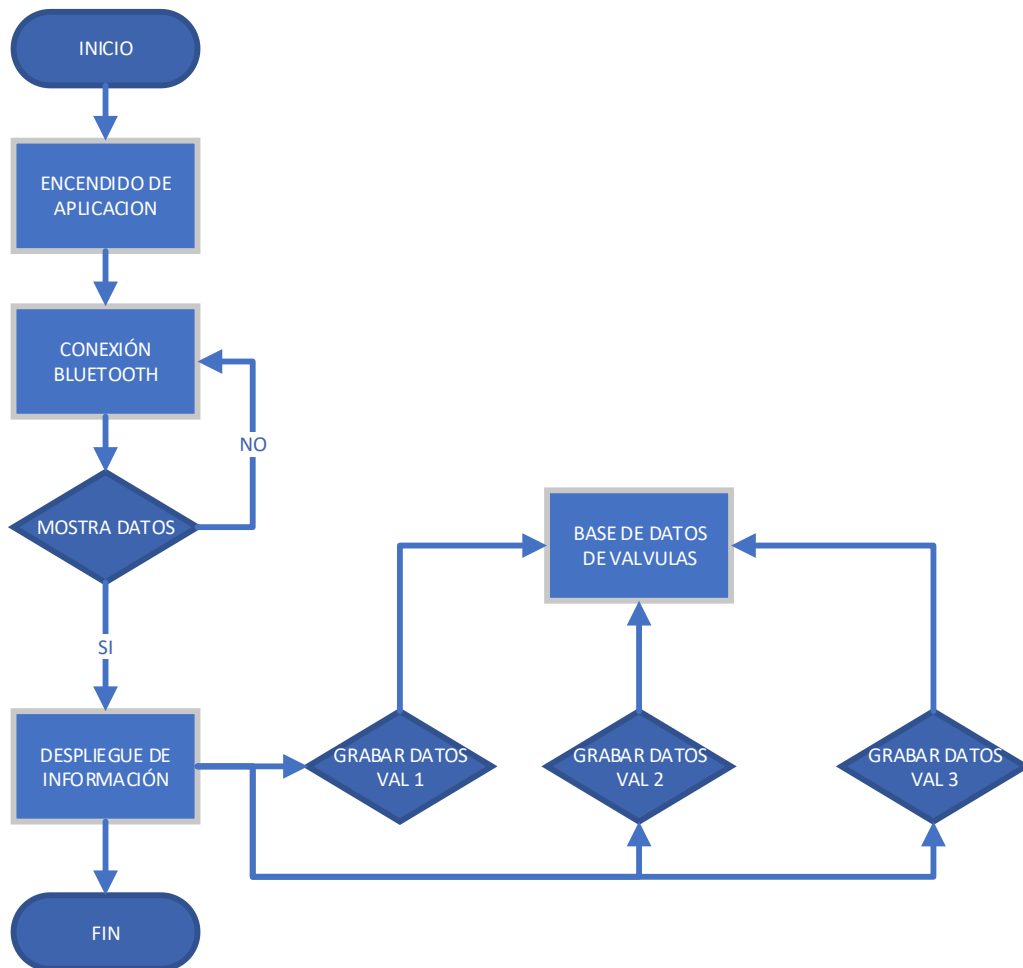


Figura. 4.26 Diagrama de flujo de visualización de parámetros configurados.

Fuente: Elaborado por el Autor

4.8.5 Diseño Lista de válvulas configuradas.

En el diseño de la lista de válvulas configuradas, se realiza un diagrama de flujo, para mejor explicación se inicia en el menú de lista válvulas configuradas a continuación, se selecciona la opción de mostrar válvulas, esta muestra los cinco registros anteriormente guardados en la base de datos, en la cual se despliegan los datos de hora actual, apertura y cierre, como se muestra en la figura 4.27.

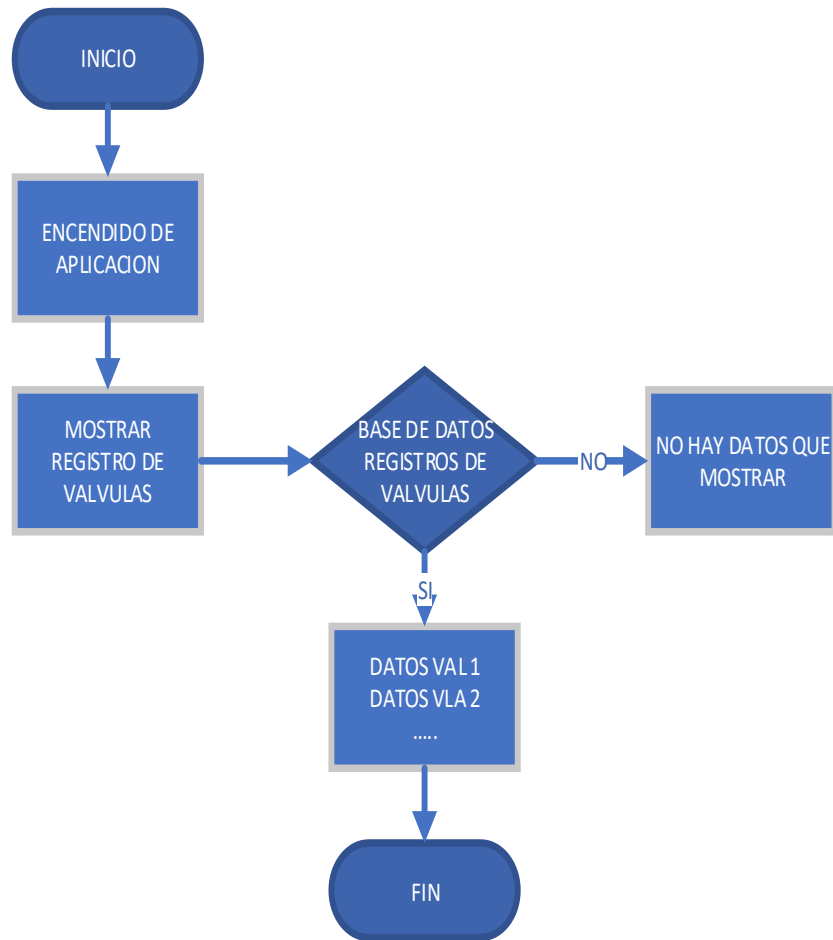


Figura. 4.27 Diagrama de flujo de visualización lista de válvulas configuradas.

Fuente: Elaborado por el Autor

4.8.6 Programación del botón y menú Lista de válvulas Configuradas.

En la siguiente figura 4.28, se indica la programación del menú de la lista de válvulas configuradas, en la cual se tiene varios registros de las válvulas anteriormente registradas y grabadas, esta visualización se la realiza con el botón Mostrar Válvulas.

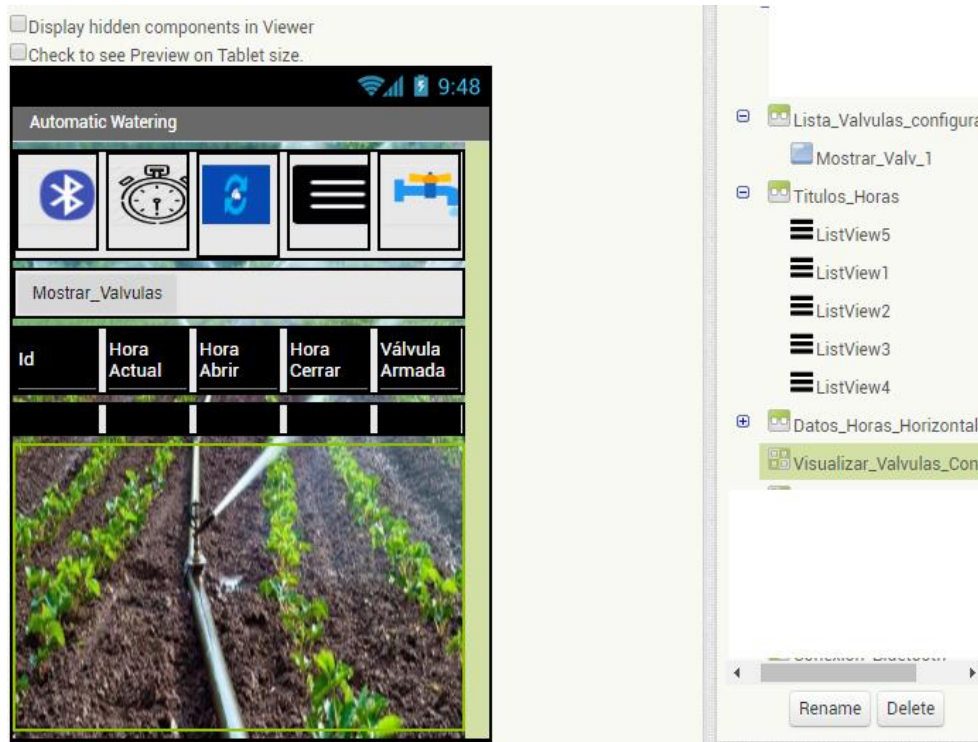


Figura. 4.28. Menú mostrar válvulas configuradas.

Fuente: Elaborada por el Autor.

En la programación del menú del botón de Mostrar Válvulas se habilita únicamente lo concerniente a dicho menú, mientras que los otros menús son deshabilitados momentáneamente.

4.8.7 Diseño de programación de apertura manual.

En el diseño del menú de apertura y cierre manual se realiza un diagrama de flujo, este comienza con la conexión *Bluetooth* con la válvula, a continuación, en el menú apertura manual se tiene las opciones manuales para abrir o cerrar la válvula de forma manual, cada una de estas opciones acciona el servomotor sin importar que se encuentre o no configurada, como se puede verificar en la figura 4.29.

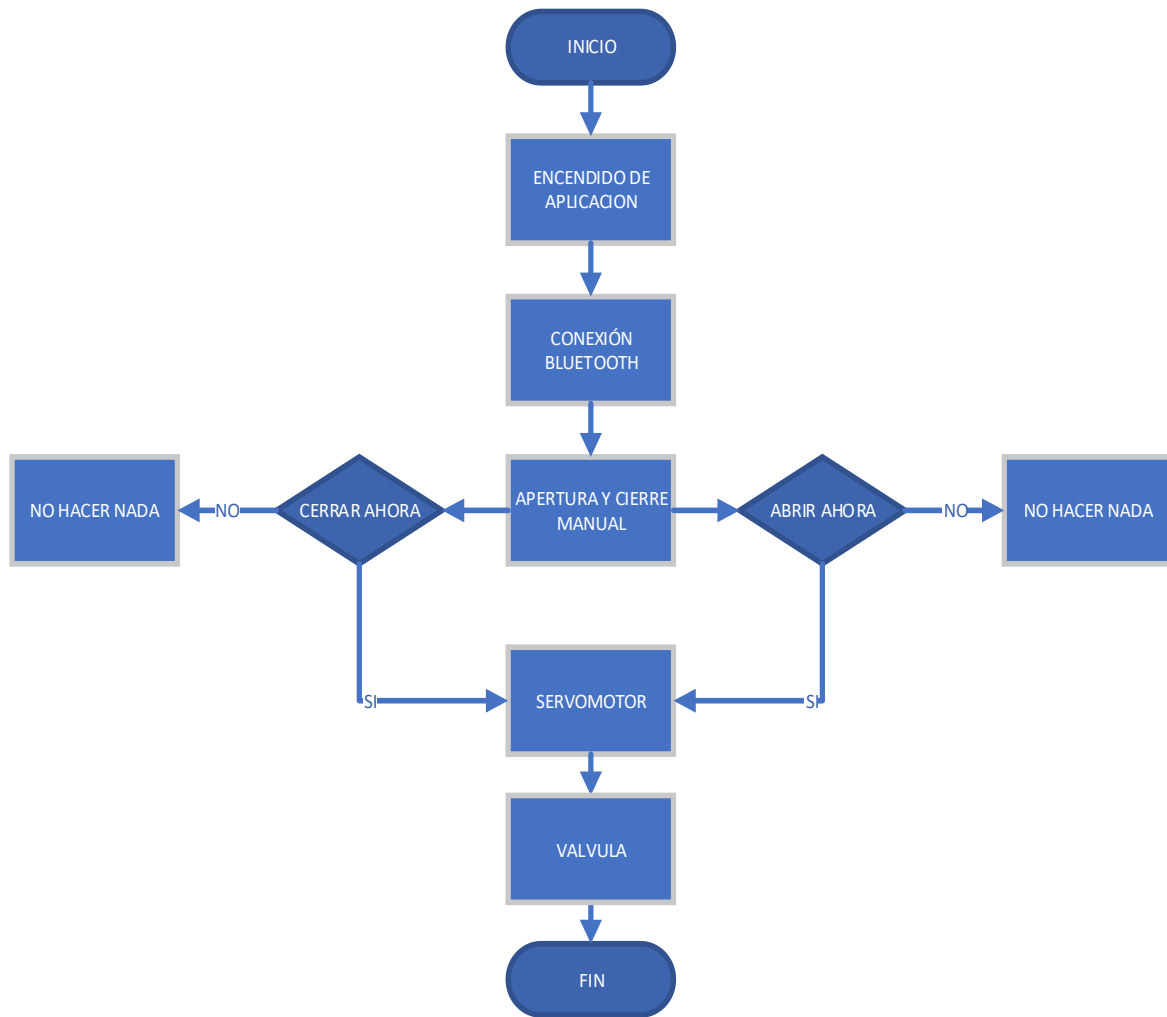


Figura. 4.29 Diagrama de flujo de visualización lista de válvulas configuradas.

Fuente: Elaborado por el Autor

En caso de presentarse algún inconveniente, se dispone de la opción de apertura manual de la válvula, con esta opción de abrir y cerrar la válvula de forma manual en el momento que se desee, así también se puede validar la configuración de los botones para la apertura y cierre de forma manual, como se detalla en la figura 4.30.



Figura. 4.30. Apertura manual

Fuente: Elaborado por el Autor

4.9 Implementación.

En la implementación del dispositivo se realiza el armado de todas las partes constitutivas de la válvula, como son: parte mecánica, electrónica y comunicaciones, en primera instancia se tiene la instalación y acople de la base inferior sobre la válvula.

Luego de la instalación de la base se prosigue a realizar el armado de todos los componentes, teniendo en cuenta la ubicación de cada una de las partes a ser instaladas.

4.9.1 Armado de la válvula.

En la figura 4.31, se puede observar la pieza que va en el cuello de la válvula, esta pieza tiene la misma forma que la válvula, la base calza de forma exacta sobre la parte metálica de la válvula, sobre esta se instala el servomotor y los soportes la base de los engranes.



Figura. 4.31. Base inferior de soporte de válvula.

Fuente: Elaborado por el Autor

En la figura 4.32, se encuentra la base inferior instalada, esta se encuentra perfectamente acoplada sobre la válvula, así también se encuentra un empaque de caucho el cual permitirá que no ingrese la humedad hacia el interior de esta.



Figura. 4.32 Montaje de la base sobre la válvula.

Fuente: Elaborado por el Autor

4.9.2 Instalación servomotor MG-995.

En las figuras 4.33, se puede apreciar el montaje del servomotor MG-995, este se encuentra perfectamente instalado en el espacio destinado para este fin.



Figura. 4.33. Instalación de servomotor MG_995.

Fuente: Elaborado por el Autor

4.9.3 Instalación tren de engranes completo.

En las figuras 4.34, 4.35 y 4.36, se puede apreciar el ensamblaje del tren de engranes,

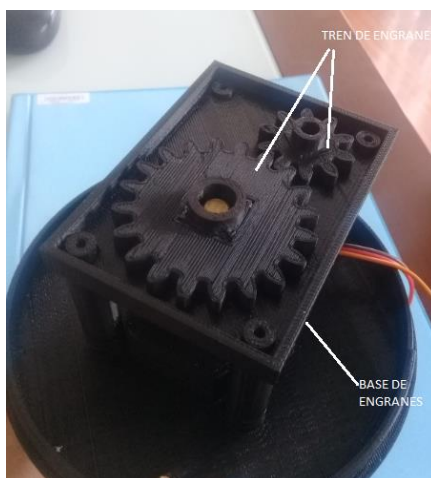


Figura. 4.34. Tren de engranes y base.

Fuente: Elaborado por el Autor.

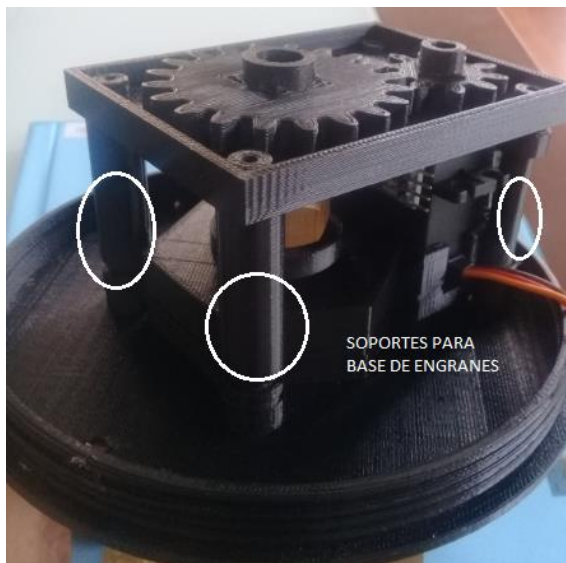


Figura. 4.35. Soportes de base de engranes.

Fuente: Elaborado por el Autor.

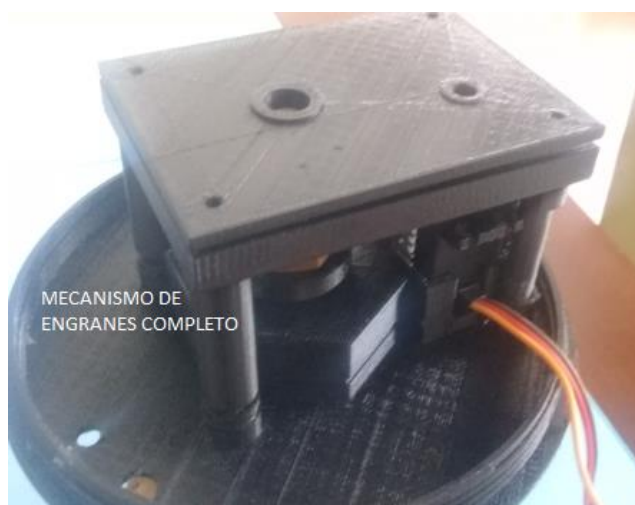


Figura. 4.36 Mecanismo de engranes completo.

Fuente: Elaborado por el Autor.

4.9.4 Instalación encoder.

En la figura 4.37, se puede apreciar la instalación del *encoder* y del sensor infrarrojo de velocidad, este se encuentra sobre el eje que hace girar el vástago de la válvula, y el sensor sobre la base del tren de engranes.

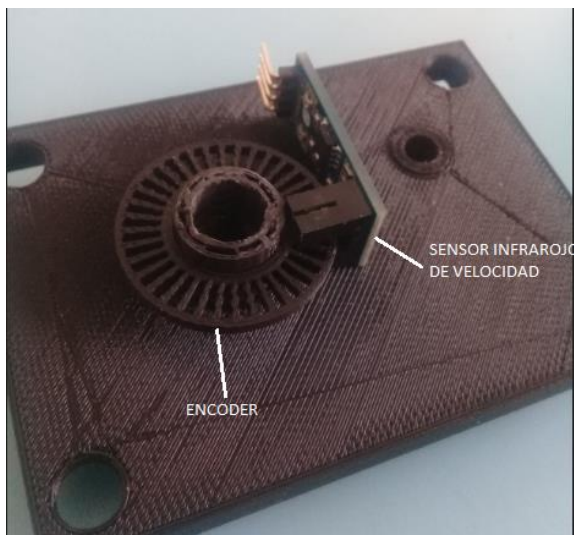


Figura. 4.37 Acople de encoder y sensor infrarrojo de velocidad.

Fuente: Elaborado por el Autor

4.9.5 Instalación Módulo *Bluetooth*

En la figura 4.38, se puede apreciar la instalación del módulo *Bluetooth*, este será el encargado de realizar la comunicación entre la válvula y el dispositivo *Smartphone*.



Figura. 4.38. Instalación módulo *Bluetooth*.

Fuente: Elaborado por el Autor.

4.9.6 Instalación del banco de baterías.

En la figura 4.39, se puede apreciar la instalación del microcontrolador Arduino Nano, de igual manera la instalación de banco de baterías y del control de las baterías.

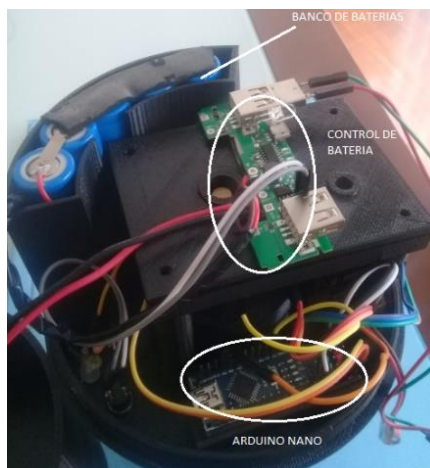


Figura. 4.39 Instalación banco de baterías y Arduino Nano.

Fuente: Elaborado por el Autor.

4.9.7 Instalación panel solar.

En la figura 4.40, se puede apreciar la instalación del panel solar, el cual proporcionará de energía para cargar el banco de baterías, este será conectado al control de baterías.

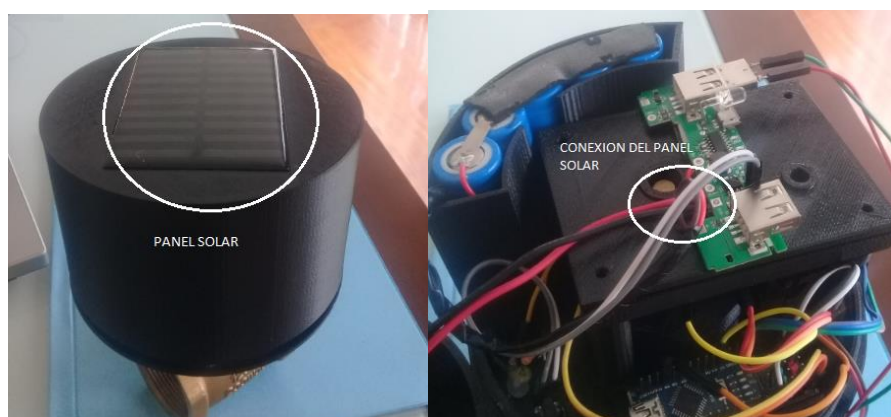


Figura. 4.40. Conexión panel solar.

Fuente: Elaborado por el Autor

4.9.8 Verificación del armado.

En la figura 4.41, se puede validar el armado de la válvula, y sus partes internas que se encuentren en la posición adecuada.

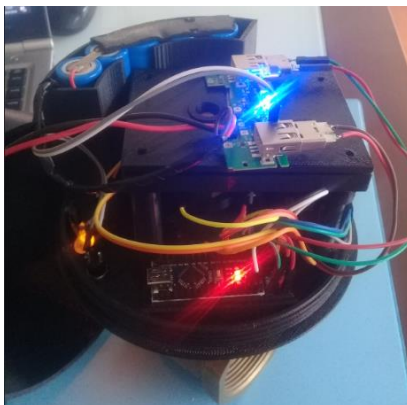


Figura. 4.41. Validación con todos sus componentes.

Fuente: Elaborado por el Autor

4.9.9 Ensamblaje completo.

En la figura 4.42, se valida que la válvula se encuentre perfectamente armado y se valida en encendido de esta.



Figura. 4.42 Armado completo de la válvula.

Fuente: Elaborado por el Autor

4.10 Pruebas de funcionamiento.

Las primeras pruebas de funcionamiento son las pruebas de encendido y apagado de la válvula, esta dispone de un interruptor de encendido y de un LED indicador, al disponer de las baterías cargadas se procede al encendido de esta, las pruebas de encendido y apagado son satisfactorias como se puede apreciar en la figura 4.43, se puede validar internamente que todos los componentes se encuentran encendidos, así como también el LED indicador.



Figura. 4.43. Pruebas de encendido y apagado de la válvula

Fuente: Elaborado por el Autor

Una vez validado el encendido de la válvula, se procede con la conexión por *Bluetooth* entre equipo *Smartphone* y el módulo *Bluetooth* de la válvula.

Como se puede observar en la figura 4.44, se realiza la conexión encuentra el equipo *Smartphone* y la válvula, se selecciona la válvula de la lista de dispositivos *Bluetooth*, luego se realiza la conexión con la válvula seleccionada, una vez conectada se procede con la configuración de esta.



Figura. 4.44. Conexión *Bluetooth* entre Smartphone y válvula.

Fuente: Elaborado por el Autor.

En la tabla 4.5, se puede evidenciar las pruebas de conectividad *Bluetooth* realizadas a diferentes distancias, en las pruebas efectuadas a distancias mayores de 10 metros existen errores de conectividad.

Análisis de pruebas de conexión Bluetooth			
Número de Intentos	Errores de conexión		
	D1 (5m)	D2 (10m)	D3(15m)
1	0	0	0
2	0	0	1
3	0	0	0
4	0	0	1
5	0	0	1

Tabla 4.5 Análisis de conexión Bluetooth

Fuente: Elaborado por el Autor.

Las pruebas realizadas de configuración de los parámetros de hora actual, apertura, cierre y armado a diferentes distancias, desde el dispositivo *Smartphone* hacia la válvula, como se puede ver en la figura 4.45.

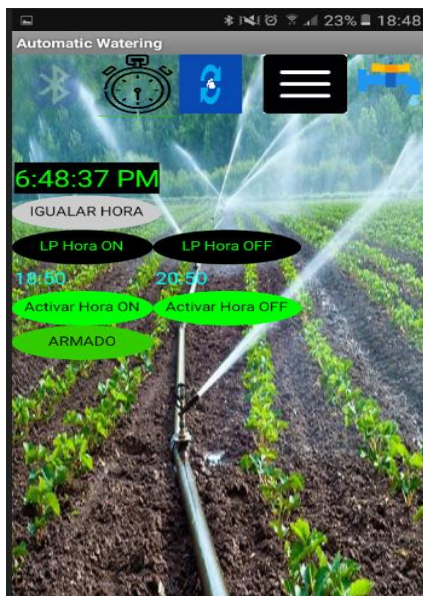


Figura. 4.45. Parámetros configurados en la válvula.

Fuente: Elaborado por el Autor.

En la tabla 4.6, se puede verificar las pruebas realizadas en la configuración de la válvula a diferentes distancias, se puede evidenciar que, a distancias mayores a 10 metros, existe errores de configuración, por motivos de conexión.

Análisis de pruebas de parámetros de configuración			
Número de Intentos	Errores de envío de datos		
	D1 (5m)	D2 (10m)	D3(15m)
1	0	0	0
2	0	0	1
3	0	1	0
4	0	0	1
5	0	0	1

Tabla 4.6 Pruebas de envío de configuración.

Fuente: Elaborado por el Autor.

En la figura 4.46, se pueden visualizar los parámetros que fueron configurados en la válvula, los datos ingresados o configurados son enviados desde la válvula hacia el equipo

Smartphone, los datos a mostrar son Hora actual, hora de apertura, hora de cierre y si la válvula se encuentra armada o no.



Figura. 4.46. Parámetros configurados y visualizados en la válvula.

Fuente: Elaborado por el Autor

En la tabla 4.7, se puede apreciar las pruebas realizadas en la recepción de los parámetros configurados, a diferentes distancias, se evidencia que a distancias mayores a las recomendadas existen errores, por motivos de conexión.

Análisis de validación de parámetros configurados			
Número de Intentos	Errores de recepción de datos		
	D1 (5m)	D2 (10m)	D3(15m)
1	0	0	0
2	0	0	1
3	0	1	0
4	0	0	1
5	0	1	1

Tabla 4.7 Pruebas de envío de configuración.

Fuente: Elaborado por el Autor.

En la figura 4.47, se valida el funcionamiento de la válvula, esta se acciona para abrir o cerrar la hora configurada, se verifica que el sensor de velocidad censa el movimiento del vástago de la válvula, en caso de no existir movimiento este envía una señal al Arduino Nado para que apague el servomotor, caso contrario el servomotor sigue funcionando.

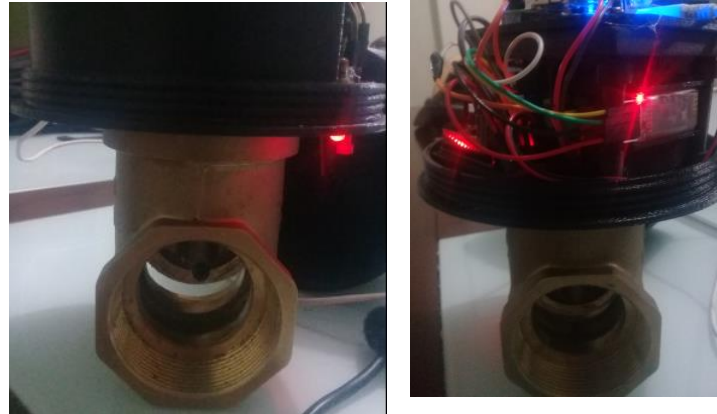


Figura. 4.47. Apertura y cierre de la valvula a la hora indicada.

Fuente: Elaborado por el Autor.

4.11 Análisis y resultados.

Luego de finalizar la construcción del prototipo de la válvula motorizada, se realiza un análisis de los resultados obtenidos de las pruebas de funcionamiento, al validar los inconvenientes presentados, con el fin de encontrar mejoras para ser implementadas en estas.

Las pruebas realizadas con el proyecto concluido dan como resultado que, al aumentar la distancia establecida por el fabricante de 10 metros, existen problemas de conexión, de igual manera para realizar las configuraciones deseadas.

Para el funcionamiento de la válvula fue necesario realizar la alimentación por separado, es decir una conexión para la parte electrónica, comunicación, y en otra conexión para el servomotor, esto se da por el circuito de control de la batería, por protección se resetea la salida USB.

El acople del tren de engranes debe ser lo más parejo posible, si estos se encuentran muy apretado o muy flojos el tren de engranes no funciona de forma correcta, y existe la posibilidad que se traben y no funcionen, por este motivo fue necesario la reubicación de los engranes a una distancia prudencial, que no estén muy flojos ni muy apretados para su funcionamiento óptimo.

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

Luego del análisis del diseño y construcción del prototipo de la válvula, se presentan las siguientes conclusiones, las que permitirán visualizar de mejor manera su funcionalidad.

- Se logró cumplir con el desarrollo de la válvula agrícola motorizada programable mediante *Bluetooth*, tanto en la parte de software como en hardware de acuerdo con las expectativas en funcionalidad, de acuerdo con las pruebas realizadas la configuración de la válvula se realiza en un tiempo promedio de un minuto, esto representa un ahorro considerable en tiempo y esfuerzo al momento de realizar la configuración de varias válvulas programables, teniendo una mejora en tiempo y esfuerzo de un 50% o más, comparadas con la instalación de válvulas normales las cuales tienen que ser manipuladas de forma manual y presencial.
- Con ayuda de los elementos electrónicos adecuados como son microcontrolador Arduino Nano, modulo Bluetooth, sensor de velocidad, servomotor, batería, se logró convertir una válvula normal de compuerta de 2 pulgadas en una válvula automatizada, programable, móvil y autónoma.
- Fue factible acoplar todas las partes constitutivas de la válvula como parte electrónica, mecánica, baterías y de comunicación en un espacio reducido, logrando de esta manera que el dispositivo sea autónomo e independiente de cualquier control externo.
- Con la utilización de un tren de engranes se pudo aumentar el torque del servomotor de 8,5 kgf-cm a 15,5 kgf-cm, gracias a relación del número de dientes de 2:1 y de sus diámetros; este torque es superior a los 4kgf-cm necesarios para mover el vástago de la válvula, el cual es utilizado sin problema para dicho fin, además el tren de

engranes fue utilizado no solo para aumentar el torque, sino también para realizar el acople entre el servomotor y el vástago de la válvula.

- Se logró realizar la comunicación y configuración de la válvula por medio de un dispositivo *Smartphone* a una distancia no mayor a 10 metros, gracias a la tecnología inalámbrica *Bluetooth*.

RECOMENDACIONES.

- Para trabajar a una distancia óptimas entre la válvula y el dispositivo *Smartphone* esta no puede ser mayor a 10 metros en línea de vista, la comunicación por *Bluetooth* no debe superar los 10 metros de acuerdo con las funcionalidades y recomendaciones del fabricante.

- Se sugiere revisar periódicamente el tren de engranes por lo menos una vez cada 3 meses dependiendo del uso de este, el trabajo y fricción de los engranes puede provocar un desgaste entre estos causando un mal funcionamiento al momento de accionar la válvula.

- Es necesario verificar que la carcasa de la válvula no sufra fuertes golpes, el armazón de la válvula está construida con material PLA, este tipo de material tiene la característica física de no tener buena resistencia a golpes extremos y caídas mayores a 50 cm sobre una base sólida.

- Se recomienda una revisión mensual del empaque de caucho que se encuentra en el cuello de la válvula, este empaque protege del ingreso de agua, humedad, polvo en el interior de esta, esta filtración puede causar problemas en el funcionamiento de sus partes electrónicas internas.

BIBLIOGRAFIA

Savant, C., Roden, M., & Carperter, G. (2000). *Diseño Electrónico, Circuitos y Sistemas*. Pearson Educación

Candela, F., & Corrales, J. (Septiembre de 2007). *Servomotores*. Obtenido de <http://www.aurova.ua.es/previo/dpi2005/docs/publicaciones/pub09-ServoMotores/servos.pdf>

Hernández, R., Fernández, C., & Batista, M. (2010). *Metodología de la Investigación*. México: McGraw Hill Educación

McRoberts, R.(2010). *Beginning Arduino*, Copyright

Torrente, O.(2013). *Arduino. Curso práctico de formación*. Primera Edición Alfaomega Grupo Editor, S.A. de C.V., México

Prabhu, C. (2004). *Bluetooth Technology*. New Delhi: PHI

Savant, C., Roden, M., & Carperter, G. (2000). *Diseño Electrónico, Circuitos y Sistemas*. Pearson Educación

Candelas,F, Corrales, J (2007) .*Servomotores*, Publicación Interna 9

Schneider, B. (s.f.). *A Guide to LiPo Batteries*. Obtenido de http://www.rcaces.org/RC%20Aces%20News%20Letter/A_Guide_to_LiPo_Batteries.pdf

Tran, P. (2017). *SolidWorks Basic Tools 2018*. SDC Publication

materialesriegos. (2013). *materialesriegos*. Obtenido de <https://www.materialesriegos.com/novedades-agricolas>.

novedades-agricolas. (2016). *novedades-agricolas*. Obtenido de <http://www.novedades-agricolas.com/es/riego/sistemas-de-riego>

phi-education. (2014). *phi-education*. Obtenido de <http://phi-education.com/store/servo-motor-MG995>

Ramírez, M. (2016). *youtube*. Obtenido de <https://www.youtube.com/watch?v=rmzfR1ok6PU>

ANEXOS

ANEXO 1: CODIGO ARDUINO

```

#include <Wire.h>
#include <TimerOne.h>
String readString;
String SELECCION;
String HORA;
String MINUTO;
String SEGUNDO;
int ind1;
int ind2;
int ind3;
int ind4;
int SegundosAbrir=0;
int SegundosCerrar=0;
char buffer[10];
int horas=0; //variable para el inicio horas
int minutos=0; //variable para el inicio minutos
volatile int segundos=0; //variable para el incremento de segundos
volatile boolean actualizar=true; //variable booleana para decicion de actualizar
volatile boolean valvula_1_on=false;
volatile boolean valvula_1_off=false;
volatile boolean AbrirManual=false;
volatile boolean CerrarManual=false;
boolean counted=false;
int t1=0,t2=0;
int hits=0;
int rps=0;
int contador_On =0;
int contador_Off =0;
#define PMS_PIN 2 // Pin for signal from Photomicrosensor
const int VALVULA = 9;
char texto[10]; //variable para el mostrar el texto de una longitud de 10 caracteres
char texto1[10];
char texto2[10];
int modo = 0; // variable modo para que inicialice con 0
const int N=16; //variable para en numero de opciones del boton mode

```

```

int horas_valvula_1_On=0; //Variable para modificar horas para abrir la valvula 1
int minutos_valvula_1_On=0; //Variable para modificar minutos para abrir la valvula 1
int segundos_valvula_1_On=0;
int horas_valvula_1_Off=0; //Variable para modificar horas para cerrar la valvula 1
int minutos_valvula_1_Off=0; //Variable para modificar minutos para cerrar la valvula 1
int segundos_valvula_1_Off=0;
void setup(){
  pinMode(VALVULA, OUTPUT);
  Serial.begin(9600);
  Wire.begin(); //activamos el LCD
  Timer1.initialize(1000000);
  Timer1.attachInterrupt(manejadoraTimer);
}
void manejadoraTimer(){ //creacion de la funcion manejadoraTimer la cual hace que inicialice el conteo de segundos
  segundos++; // comienza el tiempo en segundos
  SegundosAbrir++;
  SegundosCerrar++;
}
void actualizarReloj(){
  minutos += segundos / 60;
  segundos = segundos % 60;
  horas += minutos / 60;
  minutos = minutos % 60;
  horas = horas % 24;
}
void serialEvent() {
  while (Serial.available()) {
    char c = (char)Serial.read(); //gets one byte from serial buffer
    if (c == '*') {
      Serial.println();
      Serial.print("captured String is : ");
      Serial.println(readString); //prints string to serial port out
      ind1 = readString.indexOf(':'); //finds location of first ,
      SELECCION = readString.substring(0, ind1); //captures first data String
      ind2 = readString.indexOf(':', ind1+1 ); //finds location of second ,
    }
  }
}

```

```

HORA = readString.substring(ind1+1, ind2+1); //captures second data String
ind3 = readString.indexOf(':', ind2+1 );
MINUTO = readString.substring(ind2+1, ind3+1);
ind4 = readString.indexOf(':', ind3+1 );
SEGUNDO = readString.substring(ind3+1); //captures remain part of data after last ,
int SELEC = SELECCION.toInt();
switch (SELEC) {
case 1:
    IgualarHora();
    break;
case 2:
    IgualarHoraOn();
    brek;
case 3:
    IgualarHoraOff();
    break;
case 4:
    EnviarHora();
    break;
case 5:
    ArmarValvula();
    break;
case 6:
    ManualAbrirValvula();
    break;
case 7:
    ManualCerrarValvula();
    break;
default:
    break;
}
readString=""; //clears variable for new input
SELECCION="";
HORA="";
MINUTO="";

```

```

SEGUNDO="";
}
else {
readString += c; //makes the string readString
}
}
}

void IgualarHora() {
int x =HORA.toInt();
horas = x;
int y =MINUTO.toInt();
minutos = y;
int z =SEGUNDO.toInt();
segundos = z;
}

void IgualarHoraOn() {
int x = HORA.toInt();
horas_valvula_1_On = x;
int y = MINUTO.toInt();
minutos_valvula_1_On = y;
}

void IgualarHoraOff() {
int x = HORA.toInt();
horas_valvula_1_Off = x;
int y =MINUTO.toInt();
minutos_valvula_1_Off = y;
}

void ArmarValvula() {
valvula_1_on=true;
valvula_1_off=true;
}

void ManualAbrirValvula() {
SegundosAbrir=0;
AbrirManual=true;
}

```

```

}
void ManualCerrarValvula() {
SegundosCerrar=0;
CerrarManual=true;
}
void valvula1On(){
digitalWrite(VALVULA,HIGH);
delay(1);
digitalWrite(VALVULA,LOW);
delay(15);
}
void valvula1Off(){
digitalWrite(VALVULA,HIGH);
delay(2);
digitalWrite(VALVULA,LOW);
delay(15);
}
vod EnviarHora() {
delay(150);
Serial.print("#S1");
Serial.print(":");
Serial.println(texto);
delay(150);
Serial.print("#S2");
Serial.print(":");
Serial.println(texto1);
delay(150);
Serial.print("#S3");
Serial.print(":");
Serial.println(texto2);
delay(150);
Serial.print("#S4");
Serial.print(":");
Serial.println(texto2);
}
if ( valvula_1_on==true && valvula_1_off==true){
Serial.print("#S4");
Serial.print(":");
Serial.println(texto2);
}

```

```

}
}
void loop(){
  actualizarReloj();
  t2 = millis();
  if(t2 >= (t1 + 100)){
    rps = hits;
    hits = 0;
    t1=t2;
  }
  if(digitalRead(PMS_PIN) == HIGH){
    if(!counted){
      counted = true;
      hits++;
    }
    } else {
      counted = false;
    }
    if(AbrirManual== true){
      if(AbrirManual== true && SegundosAbrir < 23){
        if(AbrirManual==true && contador_On < 2){
          valvula1On();
        }
        if(rps > 0 ){
          valvula1On();
          contador_On++;
        }
        } else{
          digitalWrite(VALVULA,LOW);
        }
        contador_Off=0;
      }
    }
  }
}

```

```

if(CerrarManual== true){
    if(CerrarManual== true && SegundosCerrar < 23){
        if(CerrarManual==true && contador_Off < 2){
            valvula1Off();
        }
        if(rps > 0 ){
            valvula1Off();
            contador_Off++;
        }
        else{
            digitalWrite(VALVULA,LOW);
        }
        contador_On=0;
    }
}

if(valvula_1_on == true){
    if(horas == horas_valvula_1_On && minutos == minutos_valvula_1_On && segundos < 22 ){
        if(valvula_1_on==true && contador_On < 2){
            valvula1On();
        }
        if(rps > 0 ){
            valvula1On();
            contador_On++;
        }
        else{
            digitalWrite(VALVULA,LOW);
        }
        contador_Off=0;
    }
}

if(valvula_1_off == true){
    if(horas == horas_valvula_1_Off && minutos == minutos_valvula_1_Off && segundos < 22 ){
        if(valvula_1_on==true && contador_Off < 2){
            valvula1Off();
        }
    }
}

```



```
if(rps > 0 ){
  valvula1Off();
  contador_Off++;
}
else{
  digitalWrite(VALVULA,LOW);
}
contador_On=0;
}
}
Serial.println(texto);
sprintf(texto, "%02d:%02d:%02d",horas,minutos,segundos);
sprintf(texto1, "%02d:%02d:%02d",horas_valvula_1_On,minutos_valvula_1_On,segundos_valvula_1_On);
sprintf(texto2,
"%02d:%02d:%02d",horas_valvula_1_Off,minutos_valvula_1_Off,segundos_valvula_1_Off);
}
```

ANEXO 2. PROGRAMACIÓN APP INVENTOR

```
when Boton_BT .Click
do
  set Label_Conf_Valv . BackgroundColor to Menu_Principal . BackgroundColor
  set Label_Home . BackgroundColor to Menu_Principal . BackgroundColor
  set ListaValvulas . BackgroundColor to Menu_Principal . BackgroundColor
  set Label_Manual . BackgroundColor to Menu_Principal . BackgroundColor
  set Label_BT . BackgroundColor to #00FF00
  set Conexión_Bluetooth . Visible to true
  set Control_Válvula_Manual . Visible to false
  set Igualar_Hora_Sistema . Visible to false
  set Home_Principal . Visible to false
  set Muestra_Parametros_Valvula . Visible to false
  set Calibrar_Valvula . Visible to false
  set Igualar_Hora_Valvula . Visible to false
  set Lista_Valvulas_Armadas . Visible to false
  set Lista_Valvulas_configuradas . Visible to false
  set Visualizar_Valvulas_Configuradas . Visible to false
  set Titulos_Horas . Visible to false
  set Datos_Horas_Horizontal . Visible to false
```

```
initialize global MAC to "AA-AA-AA-AA-AA-AA"
```

```
when Lista_de_BT .BeforePicking
do
  set Lista_de_BT . Elements to BluetoothClient1 . AddressesAndNames
```

```
when Lista_de_BT .AfterPicking
do
  set BT_Seleccionado . Text to join trim segment text Lista_de_BT . Selection
  start 18
  length length Lista_de_BT . Selection - 17
  set Conectado_BT . Text to "NO CONECTADO"
  set global MAC to segment text Lista_de_BT . Selection
  start 1
  length 17
```

```
when Conectar_BT . Click
do
  set Lista_de_BT . Selection to call BluetoothClient1 . Connect
  address get global MAC
  if BluetoothClient1 . IsConnected
  then
    set Conectado_BT . Text to " CONECTADO "
    if BT_Seleccionado . Text = " WWW "
    then
      set Mostrar_Valvula . Text to BT_Seleccionado . Text
    else
      set Conectado_BT . Text to " ERROR CONECCION "
```

```
when Desconectar_Valvula . Click
do
  call BluetoothClient1 . Disconnect
  set Conectado_BT . Text to " "
  set BT_Seleccionado . Text to " "
```

```
when Boton_Conf_Valv . Click
do
  set Label_BT . BackgroundColor to Menu_Principal . BackgroundColor
  set ListaValvulas . BackgroundColor to Menu_Principal . BackgroundColor
  set Label_Manual . BackgroundColor to Menu_Principal . BackgroundColor
  set Label_Home . BackgroundColor to Menu_Principal . BackgroundColor
  set Label_Conf_Valv . BackgroundColor to #00FF00
  set Igualar_Hora_Sistema . Visible to true
  set Igualar_Hora_Valvula . Visible to true
  set Conexión_Bluetooth . Visible to false
  set Home_Principal . Visible to false
  set Control_Válvula_Manual . Visible to false
  set Muestra_Parametros_Valvula . Visible to false
  set Lista_Valvulas_Armadas . Visible to false
  set Calibrar_Valvula . Visible to false
  set Lista_Valvulas_configuradas . Visible to false
  set Visualizar_Valvulas_Configuradas . Visible to false
  set Titulos_Horas . Visible to false
  set Datos_Horas_Horizontal . Visible to false
```

```
when IGUALAR_HORA .Click
do call BluetoothClient1 .SendText
   text join "1"
   "0"
   call Clock1 .Hour
   instant call Clock1 .Now
   "0"
   call Clock1 .Minute
   instant call Clock1 .Now
   "0"
   call Clock1 .Second
   instant call Clock1 .Now
   "*"

```

```
when LP_Hora_ON .AfterTimeSet
do set Label_Hora_ON .Text to join LP_Hora_ON .Hour
  "0"
  LP_Hora_ON .Minute

```

Show Warnings

```
when LP_Hora_ON .AfterTimeSet
do set Label_Hora_ON .Text to join LP_Hora_ON .Hour
  "0"
  LP_Hora_ON .Minute

```

```
when LP_Hora_OFF .AfterTimeSet
do set Label_Hora_OFF .Text to join LP_Hora_OFF .Hour
  "0"
  LP_Hora_OFF .Minute

```

```
when Activar_Valvula_OFF .Click
do call BluetoothClient1 .SendText
   text join "3"
   "0"
   Label_Hora_OFF .Text
   "0"
   "00"
   "*"

```

```
when Armar_Valvula .Click
do call BluetoothClient1 .SendText
   text join "5"
   "*"
   set Armar_Valvula .Text to "ARMADO"
   set Armar_Valvula .BackgroundColor to

```

```
when Activar_Valvula_ON .Click
do
  call BluetoothClient1 .SendText
  text join "2"
  "0"
  Label_Hora_ON .Text
  "0"
  "00"
  "*"

```

```
when Boton_Home .Click
do
  set Label_BT .BackgroundColor to Menu_Principal .BackgroundColor
  set ListaValvulas .BackgroundColor to Menu_Principal .BackgroundColor
  set Label_Manual .BackgroundColor to Menu_Principal .BackgroundColor
  set Label_Conf_Valv .BackgroundColor to Menu_Principal .BackgroundColor
  set Label_Home .BackgroundColor to #00FF00
  set Igualar_Hora_Sistema .Visible to false
  set Conexión_Bluetooth .Visible to false
  set Control_Válvula_Manual .Visible to false
  set Home_Principal .Visible to true
  set Muestra_Parametros_Valvula .Visible to true
  set Calibrar_Valvula .Visible to false
  set Igualar_Hora_Valvula .Visible to false
  set Lista_Valvulas_Armadas .Visible to true
  set Lista_Valvulas_configuradas .Visible to false
  set Visualizar_Valvulas_Configuradas .Visible to false
  set Titulos_Horas .Visible to false
  set Datos_Horas_Horizontal .Visible to false

```

```
when Salir_Aplicación .Click
do
  call BluetoothClient1 .Disconnect
  close application

```

```

when Clock_Read -> .Timer
do
  if BluetoothClient1 -> .IsConnected
  then
    initialize local mensaje to call BluetoothClient1 -> .ReceiveText
    numberOfBytes call BluetoothClient1 -> .BytesAvailableToReceive
    in
      if starts at text get mensaje -> [1]
      piece "#S"
      then
        initialize local DatosSensor to call LeerSensor ->
        mensaje segment text get mensaje ->
        start [3]
        length length get mensaje -> - [2]

        initialize local numSensor to [1]
        initialize local Hora to [1]
        initialize local Minuto to [1]
        initialize local Segundo to [1]
        in
          set numSensor to select list item list get DatosSensor ->
          index [1]
          set Hora to select list item list get DatosSensor ->
          index [2]
          set Minuto to select list item list get DatosSensor ->
          index [3]
          set Segundo to select list item list get DatosSensor ->
          index [4]
          call ColocarDatos ->
          sensor get numSensor ->
          hora get Hora ->
          minuto get Minuto ->
          segundo get Segundo ->
  }
}

```

```

initialize global List_Id to [ ]
initialize global List_Hora to [ ]
initialize global List_Abrir to [ ]
initialize global List_Cerrar to [ ]
initialize global List_Armar to [ ]
initialize global List_Cont to [0]

```

```

when Mostrar_Valv_1 -> .Click
do
  set ListId -> .Elements to get global List_Id ->
  set List_Hora_Actual -> .Elements to get global List_Hora ->
  set List_Hora_Abrir -> .Elements to get global List_Abrir ->
  set List_Hora_Cerrar -> .Elements to get global List_Cerrar ->
  set List_Armar -> .Elements to get global List_Armar ->
  call DB_1 -> .StoreValue
  tag List_Cont ->
  valueToStore get global List_Cont ->
  call DB_1 -> .StoreValue
  tag List_Id ->
  valueToStore get global List_Id ->
  call DB_1 -> .StoreValue
  tag List_Hora ->
  valueToStore get global List_Hora ->
  call DB_1 -> .StoreValue
  tag List_Abrir ->
  valueToStore get global List_Abrir ->
  call DB_1 -> .StoreValue
  tag List_Cerrar ->
  valueToStore get global List_Cerrar ->
  call DB_1 -> .StoreValue
  tag List_Armar ->
  valueToStore get global List_Armar ->

```

```
when Screen1.Initialize
do
  set BluetoothClient1.DelimiterByte to 10
  if not BluetoothClient1.Enabled
  then call ActivityStarter1.StartActivity

  set global List_Id to call DB_1.GetValue
  tag List_Cont
  valueIfTagNotThere 0

  set global List_Id to call DB_1.GetValue
  tag List_Id
  valueIfTagNotThere create empty list

  set global List_Hora to call DB_1.GetValue
  tag List_Hora
  valueIfTagNotThere create empty list

  set global List_Abrir to call DB_1.GetValue
  tag List_Abrir
  valueIfTagNotThere create empty list

  set global List_Cerrar to call DB_1.GetValue
  tag List_Cerrar
  valueIfTagNotThere create empty list

  set global List_Armar to call DB_1.GetValue
  tag List_Armar
  valueIfTagNotThere create empty list
```

```
when Lista_Valvulas_Totales.Click
do
  set Label_BT.BackgroundColor to Menu_Principal.BackgroundColor
  set Label_Home.BackgroundColor to Menu_Principal.BackgroundColor
  set Label_Manual.BackgroundColor to Menu_Principal.BackgroundColor
  set Label_Conf_Valv.BackgroundColor to Menu_Principal.BackgroundColor
  set ListaValvulas.BackgroundColor to #000000
  set Igualar_Hora_Sistema.Visible to false
  set Conexión_Bluetooth.Visible to false
  set Control_Valvula_Manual.Visible to false
  set Home_Principal.Visible to false
  set Muestra_Parametros_Valvula.Visible to false
  set Lista_Valvulas_configuradas.Visible to true
  set Home_Principal.Visible to false
  set Muestra_Parametros_Valvula.Visible to false
  set Lista_Valvulas_Armadas.Visible to false
  set Visualizar_Valvulas_Configuradas.Visible to true
  set Titulos_Horas.Visible to true
  set Datos_Horas_Horizontal.Visible to true
```



```

to ColocarDatos sensor hora minuto segundo
do
  if get sensor = 1
  then set HoraActual . Text to join get hora
    . get minuto
    . get segundo
  else if get sensor = 2
  then set HoraAbrir . Text to join get hora
    . get minuto
    . get segundo
  else if get sensor = 3
  then set HoraCerrar . Text to join get hora
    . get minuto
    . get segundo
  else if get sensor = 4
  then set Armada . Text to join "ARMADA"

```

```

when Prg_Val_1 . Click
do
  add items to list list get global List_Id
  item 1
  add items to list list get global List_Hora
  item HoraActual . Text
  add items to list list get global List_Abrir
  item HoraAbrir . Text
  add items to list list get global List_Cerrar
  item HoraCerrar . Text
  add items to list list get global List_Armar
  item Armada . Text
  set global List_Cont to 1

```

```

when Prg_Val_2 . Click
do
  add items to list list get global List_Id
  item 2
  add items to list list get global List_Hora
  item HoraActual . Text
  add items to list list get global List_Abrir
  item HoraAbrir . Text
  add items to list list get global List_Cerrar
  item HoraCerrar . Text
  add items to list list get global List_Armar
  item Armada . Text
  set global List_Cont to 2

```

```

when Prg_Val_3 . Click
do
  add items to list list User tapped and released the button.
  item 3
  add items to list list get global List_Hora
  item HoraActual . Text
  add items to list list get global List_Abrir
  item HoraAbrir . Text
  add items to list list get global List_Cerrar
  item HoraCerrar . Text
  add items to list list get global List_Armar
  item Armada . Text
  set global List_Cont to 3

```

```

when Prg_Val_4 . Click
do
  add items to list list get global List_Id
  item 4
  add items to list list get global List_Hora
  item HoraActual . Text
  add items to list list get global List_Abrir
  item HoraAbrir . Text
  add items to list list get global List_Cerrar
  item HoraCerrar . Text
  add items to list list get global List_Armar
  item Armada . Text
  set global List_Cont to 4

```

```

when Prg_Val_5 . Click
do
  add items to list list get global List_Id
  item 5
  add items to list list get global List_Hora
  item HoraActual . Text
  add items to list list get global List_Abrir
  item HoraAbrir . Text
  add items to list list get global List_Cerrar
  item HoraCerrar . Text
  add items to list list get global List_Armar
  item Armada . Text
  set global List_Cont to 5

```

ANEXO 3. ARMADO DE VALVULA

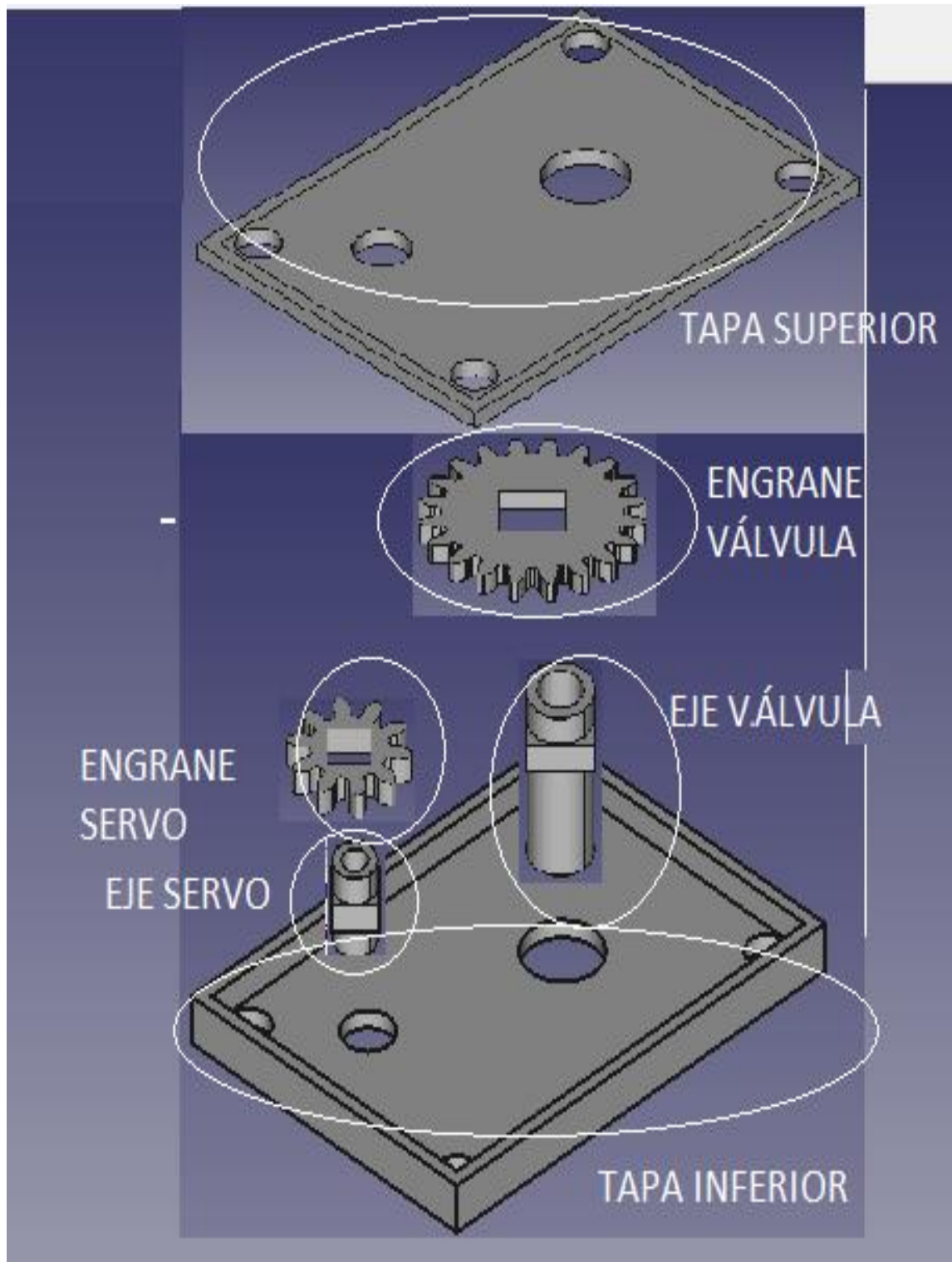


Tabla 1. Armado caja de engranes

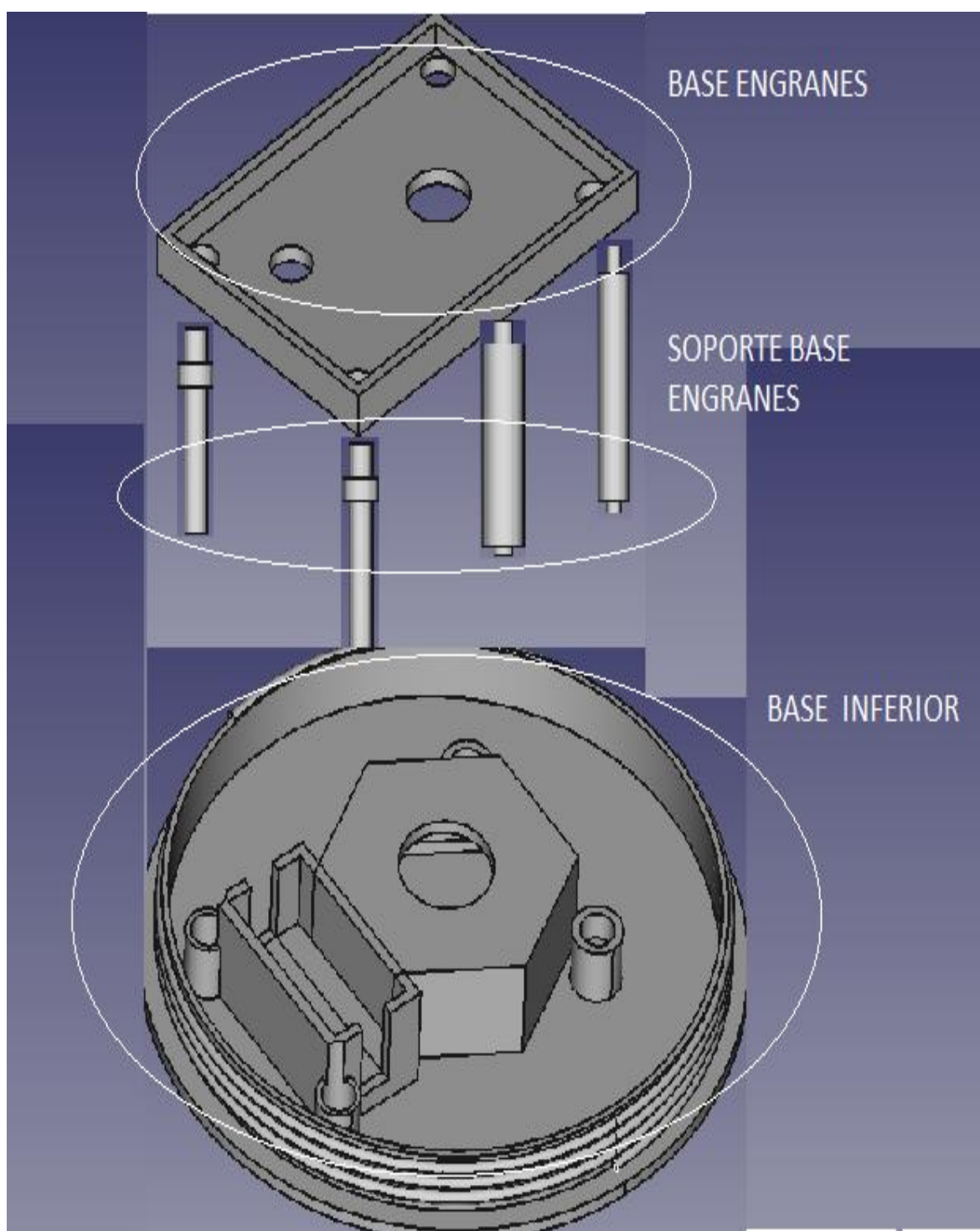


Tabla 2. Armado base con caja de engranes

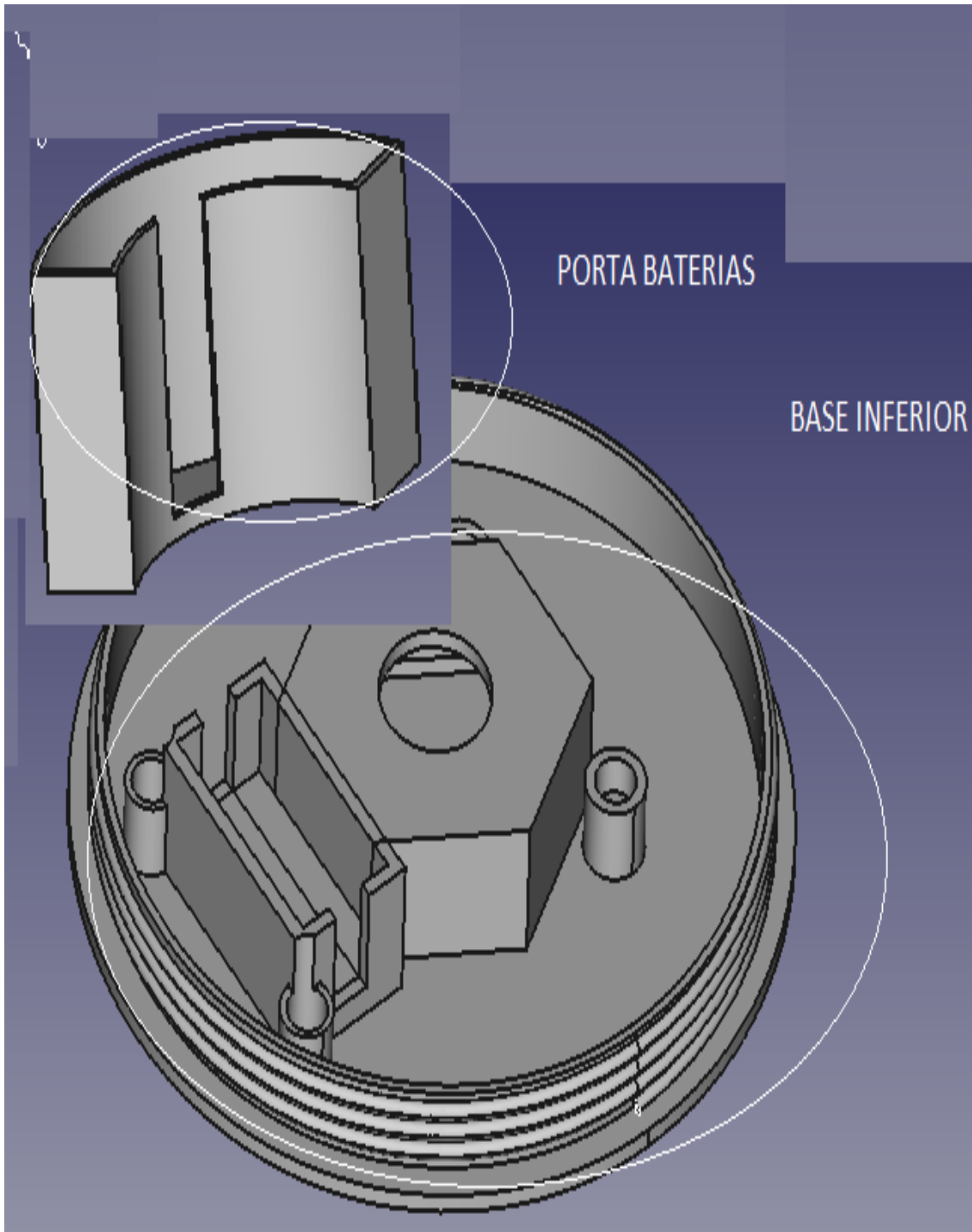


Tabla 3. Armado base con soporte de baterías

ANEXO 4. MANUAL DE USUARIO.



MANUAL DE USUARIO

VÁLVULA DE RIEGO AGRÍCOLA
MOTORIZADA PROGRAMABLE MEDIANTE
BLUETOOTH, AUTÓNOMA Y MOVIL

Descripción.

Una válvula de riego programable motorizada fue desarrollada para ayudar las labores agrícolas en los lugares alejados de la ciudad, donde no se dispone de energía eléctrica para la implementación de sistemas de riego, que funcionan con energía eléctrica de la red.

Esta válvula, está diseñada para trabajar de forma independiente sin la necesidad de un control principal, esta trabaja de forma autónoma y es programable por medio de un dispositivo *Smartphone* con su respectivo aplicativo.

Este dispositivo está diseñado para ser colocado en las instalaciones de riego, es decir para ser colocado sobre un aspersor o en su defecto en un ramal de riego.

Esta válvula, está diseñada para todo tipo de personas que tengan conocimientos básicos del manejo de un dispositivo *Smartphone*.

Objetivos del manual de usuario.

El objetivo principal del desarrollo de la válvula es ayudar a las tareas agrícolas en la parte de automatización de los riegos, es decir el dispositivo será instalado en lugares donde no existe energía eléctrica de red, ya que este trabaja con baterías internas y un panel solar con lo cual se tendrá una autonomía.

Componentes principales del prototipo.

En la figura 0.1, se muestra los componentes internos de la válvula, cada uno de estos realiza un rol importante en el funcionamiento de esta.

- Batería. - Son las encargadas de suministrar la energía para el funcionamiento de sus componentes electrónicos.
- Arduino Nano. - Es el encargado de controlar el funcionamiento de la válvula.

- Sensor de velocidad. - El sensor verificará el movimiento del vástago de la válvula.
- Módulo *Bluetooth*. - Encargado de realizar la conexión de comunicación con el dispositivo *Smartphone*.
- Mecanismo de engranes. - Encargados de hacer girar el vástago de la válvula.
- Servomotor MG-995.- Encargado de hacer girar el vástago de la válvula en el tiempo indicado.
- Control de batería. - Verifica y entrega la corriente hacia cada uno de los componentes electrónicos de la válvula.



Figura. 0.1 Partes internas de la válvula

Requerimientos para el uso del dispositivo.

- Disponer de un dispositivo *Smartphone* con sistema operativo *Android*.
- Tener instalado el aplicativo en el dispositivo *Smartphone* y disponer de conexión *Bluetooth*.

- Tener cargado las baterías internas de la válvula.

Manejo del aplicativo.

En la siguiente figura 0.2 se conecta con la válvula a ser configurada, seleccionamos la válvula y seleccionamos conectar.



Figura. 0.2 Conexión *Bluetooth*

Una vez conectada la válvula procedemos a configurar los parámetros, de hora actual del dispositivo, hora de apertura, hora de cierre y armado de la válvula.

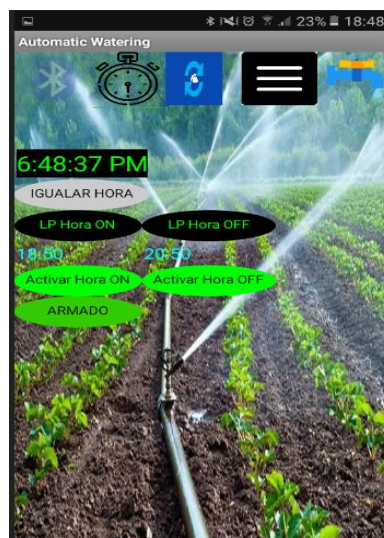


Figura. 3 Configuración de válvula

En la figura 0.3 se puede visualizar los parámetros que fueron configurados en la válvula, los datos ingresados o configurados son enviados desde la válvula hacia el equipo *Smartphone*, los datos a mostrar son Hora actual, hora de apertura, hora de cierre y si la válvula se encuentra armada o no.



Figura. 0.3. Verificación de parámetros configurados

Solución de problemas.

El Equipo no enciende. Verificamos si las baterías se encuentran cargadas y conectadas, además verificamos el interruptor de encendido.

El Equipo no se conecta por *Bluetooth*. Verificamos que el equipo este previamente configurado en el dispositivo *Smartphone* y que la distancia no supere los 10 metros.

El Equipo no se configura. Verificamos que exista conectividad entre el dispositivo *Smartphone* y la válvula.

El Equipo no abre o cierra la válvula. Verificamos que las baterías no se encuentren descargadas y que tengas suficiente carga para mover el servomotor.

Contacto de soporte Técnico

Correo electrónico

Juan Méndez juancamaneey@hotmail.com

Teléfono

Juan Méndez +593 996231497

ANEXO 5. MANUAL TÉCNICO



MANUAL TÉCNICO

VÁLVULA DE RIEGO AGRÍCOLA
MOTORIZADA PROGRAMABLE MEDIANTE
BLUETOOTH, AUTÓNOMA Y MOVIL

Objetivos del manual de técnico.

El objetivo de este manual es dar a conocer los elementos constitutivos que conforman este prototipo y su funcionamiento, guiar a la persona encargada del mantenimiento y soporte a resolver las fallas de hardware, software más comunes y simples que se pueden presentar durante el uso del prototipo, se darán indicaciones de cómo realizar un mantenimiento preventivo y correctivo de este.

Especificaciones técnicas de los elementos constitutivos.

Arduino Nano

Arduino Nano, en un pequeño controlador, completo y fácil de usar basado en *ATmega328* (Arduino nado 3.0), este es semejante en funcionalidades que el Arduino uno, pero de diferente tamaño. Este no dispone de una alimentación para voltaje DC, trabaja con un puerto Mini-b USB. En la figura 1 se puede apreciar la distribución de pines del microcontrolador Arduino Nano.



Figura. 1 Arduino Nano.

Fuente de poder

Arduino Nano puede ser encendido con un puerto Mini B USB, además de soporta entre 6-12 V DC con una fuente DC externa por el pin 30.

Memoria.

El chip *ATmega 328* tiene 32 KB, de los cuales 2KB son usados para el arranque del *bootloader*, además dispone de 2Kb de memoria RAM y 1KB de memoria EEPROM.

Entradas y salidas.

Dispone de 14 pines digitales que son usados como pines de entradas o salida, estos operan con 5VDC, cada pin puede recibir o suministrar un máximo de 40 mA, además de disponer algunos pines extras que tienen funciones especiales.

Arduino Nano dispone de 8 entradas analógicas con una resolución de 10 bits de resolución, por defecto de 0 a 5 voltios.

Comunicación.

Arduino Nano tiene la facilidad de comunicarse con un computador por medio de un puerto serial, el *ATmega328* provee *UART TTL (5V)* puerto serial de comunicación, está disponible en los puertos digitales (*RX*) (*TX*) para transmisión inalámbrica, el software de Arduino provee un puerto virtual con la computadora para recibir y enviar datos en texto plano desde el Arduino y el computador.

Programación

Los microcontroladores Arduino son programados con el software de Arduino, que es obtenido desde la página web del fabricante, este software es totalmente gratuito.

Características Arduino Nano.

El microcontrolador Arduino Nano, tiene las siguientes características, y se mencionan en la tabla 1.

<i>características</i>	
Microcontrolador:	<i>Atmega 328</i>
Voltaje de alimentación de entrada:	5V
Pines digitales:	14 (6 con PWM)
Puertos analógicos:	16
Corriente máxima suministrada por pin:	40mA
Corriente máxima suministrada para el pin 3.3V:	50 mA
Memoria flash:	32 kB
Memoria SRAM:	8 kB
Memoria EEPROM:	2 kB
Velocidad de reloj:	16 MHz

Tabla 1. Características Arduino Nano.

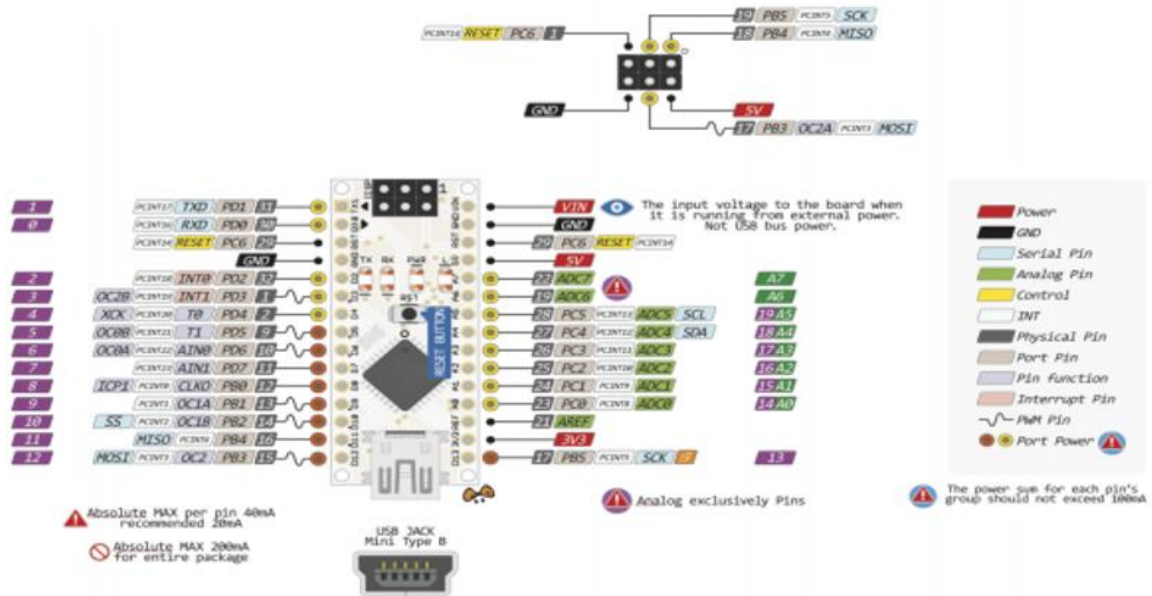


Figura. 2. Distribución de pines Arduino Nano.

Servomotor MG-995.

Los servomotores son motores electrónicos que mueven o empujan partes de máquinas con gran precisión, los servos se los encuentra en muchos lugares o artefactos electrónicos, juguetes, etc.

El diseño de los engranes hacen que la gran velocidad y bajo torque del motor sea convertida en una velocidad más baja y con un torque más grande, haciendo que el trabajo final del motor sea el mismo.

El Servomotor MG-995 dispone de una modulación digital con un gran torque el cual está comprendido entre 9,40 kg-cm con un voltaje de 4,8 VDC hasta un máximo de 11 kg-cm con 6 VDC, además dispone de un sistema de engranes de metal el cual está diseñado para trabajo continuo y de gran torque, como se observa en la figura 3.

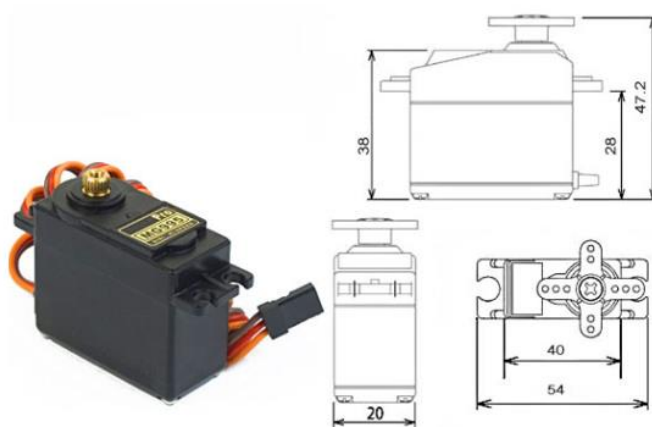


Figura. 3 servo motor MG-995.

Principales características:

- Modelo: MG-995
- Tamaño: 40.7, 19.7, 42.9mm
- Peso: 55g
- Torque 15KG-cm
- Velocidad: 53-62 RPM.
- Temperatura de operación -30 hasta 60 °
- Tipo de conector: JR, FUTABA General
- Angulo de rotación: 180 grados (truncado para giro continuo).
- Tipo: Servo
- Corriente de operación: 100mA
- Voltaje de operación: 3-7.2V

Módulo *Bluetooth* HC-05

El módulo *Bluetooth* HC-05, es utilizado en todo tipo de proyectos donde se necesite una conexión inalámbrica fiable, sencilla, económica y de corto alcance. Estos módulos se configuran mediante comandos AT y se tiene la posibilidad de hacerlos funcionar en modo

maestro o esclavo, estos módulos por defecto se encuentran en modo esclavo el cual permite conexiones entrantes.

El módulo *Bluetooth* HC-05 puede ser energizado con un voltaje de 3.3 y 6V (por defecto con voltaje de 5VDC), los pines *TX* y *RX* utilizan niveles de voltaje de 3,3V por lo que no se puede conectar directamente a placas Arduino de 5VDC, el modelo HC-05 es un módulo *Bluetooth* muy versátil, económico y fácil de usar como se puede apreciar en la figura 5



Figura. 4. Módulo *Bluetooth* HC-05.

Principales características:

- Protocolo de comunicación *Bluetooth*: v1.1 / 2.0.
- Frecuencia de trabajo: banda *ISM* de 2,4 GHz.
- Modulación de trabajo: *GFSK*
- La potencia de transmisión es de menos de 4dBm, Clase 2.
- La sensibilidad es de: Menos de -84dBm en el 0,1% BER.
- Velocidad de transmisión asíncrona de: 2.1Mbps (Max) / 160 kbps.
- Síncrono: 1Mbps / 1Mbps.
- Modo de trabajo: *Bluetooth* (maestro y esclavo).

- Fuente de alimentación DC: + 3.3VDC 50mA. (soporta de 3.3 a 6VDC)

Módulo sensor de velocidad

El sensor de velocidad está compuesto transmisor y receptor por infrarrojos y un encoder ranurado, el cual genera una serie de impulsos digitales que puede ser utilizados como sensor de velocidad con la programación adecuada.

Los codificadores ópticos están constituidos por un disco transparente o discos ranurados con una serie de marcas opacas colocadas radialmente y equidistantes, el sensor consta de un LED infrarrojo y un receptor los cuales detectan el paso de luz que atraviesa el encoder ranurado, estos pulsos son detectados y contabilizados, se puede observar el sensor en la figura 5.



Figura. 5 Módulo de velocidad FC-03

Principales características.

- Voltaje de trabajo: 3,3 a 12 V
- Chip principal: LM393
- Salida efectiva: Nivel bajo
- LEDs indicadores de funcionamiento
- Salida: Conmutación digital (0 y 1)
- Cuenta y detecta la velocidad de un motor

- Número de pines: 4
- Compatible con Arduino
- Dimensiones: 31 mm X 14.5 mm
- Modelo: FC-03

Sistema de engranajes.

Los sistemas mecánicos de engranes permiten transmitir un movimiento giratorio, velocidad, torque de un engrane a otro, se puede modificar la velocidad de giro, sentido de giro, y torque, ver figura 6.

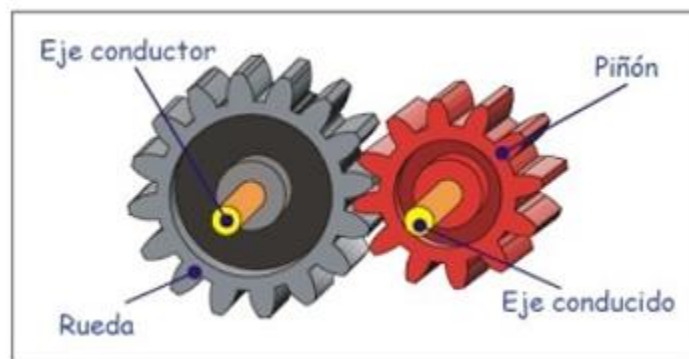


Figura. 6 Sistema de engranajes,

Un sistema de tren de engranes simple consta de una rueda motriz, con una cantidad de dientes en su periferia exterior que engrana sobre otro engrane de características similares, lo que evita el deslizamiento entre las ruedas, los engranajes de menor tamaño se llaman piñón y el de mayor diámetro se llama rueda.

Relación de transmisión.

Para el cálculo de la relación de transmisión entre engranes es necesario disponer de la cantidad de dientes, o también del diámetro de estos, la relación del número de dientes de la rueda conductora con relación al número de dientes de la rueda conducida.

$$\text{Relación de velocidades} = \frac{\text{Número de dientes de la rueda conductora}}{\text{Número de dientes de la rueda conducida}} = \frac{Z_1}{Z_2}$$

La utilidad de este tipo de mecanismos de engranes es muy amplia, estos pueden ser utilizado en la industria como reductores de velocidad y duplicadores de fuerza, como es el caso de máquinas eléctricas, mecánicas y la mayoría de los electrodomésticos que necesitan reducir la velocidad y que disponga de fuerza.

En la figura 7y 8, se muestra las partes constitutivas de esta, y un diagrama de bloques.



Figura. 7. Parte interna de la válvula

Diagrama de elementos y conexión.

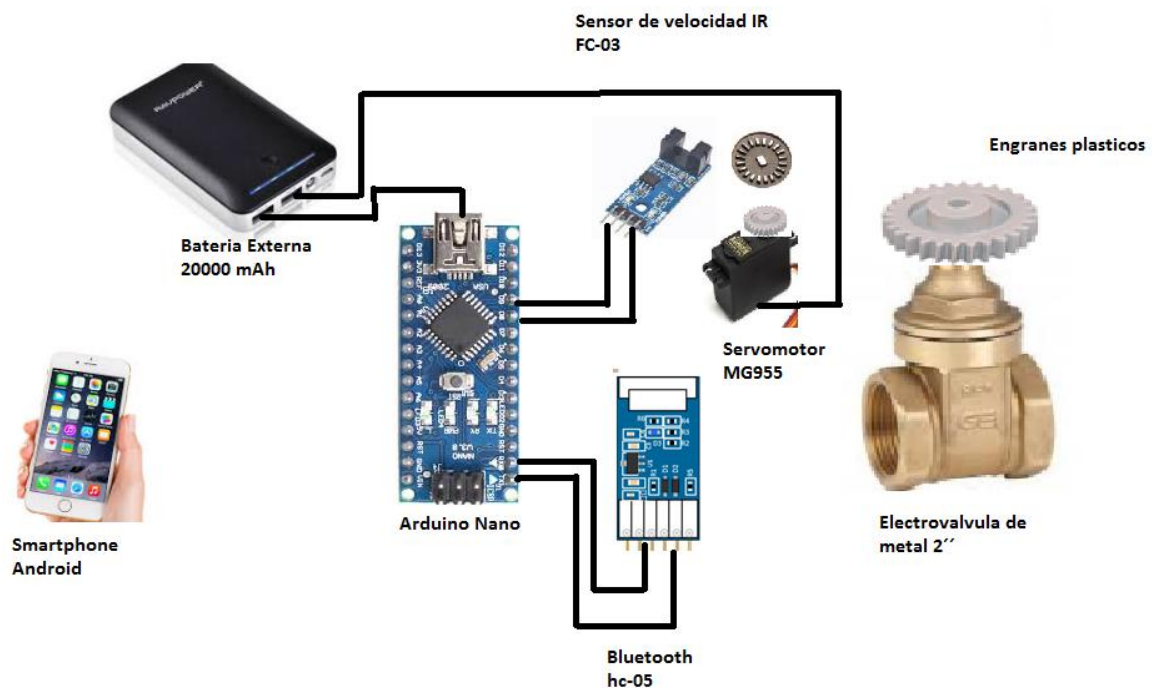


Figura. 8. Diagrama de conexión de elementos,

Solución de problemas más comunes.

El equipo no enciende.

- Validar que las baterías se encuentren con carga y conectadas.
- Verificar que los conectores USB estén conectados en el control de baterías.
- Verificar que se encuentre encendido el interruptor.

El equipo no se conecta por *Bluetooth*.

- Validar que el equipo se encuentre encendido.
- Verificar que la distancia de conexión no supere los 10 metros.
- Verificar que el módulo se encuentre conectado y operativo.

- Validar que el microcontrolador Arduino Nano funcione de forma correcta.

El equipo no se puede configurar.

- Validar que el dispositivo esté conectado por *Bluetooth*
- Verificar que no esté conectado más de 10 metros.
- Verificar que el dispositivo Smartphone tenga encendido el *Bluetooth*
- Validar los componentes internos que funcionen, Arduino Nano y Módulo *Bluetooth*.

El equipo no abre ni cierra la válvula.

- Verificar que los parámetros se encuentren configurados.
- Validar que exista suficiente carga en las baterías para encender el servomotor.
- Verificar que el servomotor esté trabaja correctamente es decir que no esté quemado.
- Verificar que el sensor infrarrojo de velocidad esté operando de forma correcta.
- Validar que los engranes no se encuentren trabados o dañados.

El equipo se prende, pero no hace nada

- Verificar que el microcontrolador Arduino Nano no se encuentre quemado.
- Cargar nuevamente el código el microcontrolador Arduino Nano.
- Verificar las baterías se encuentren con carga, medir voltajes y corrientes.
- Validar que la información de configuración este siendo ingresada en l válvula.

El equipo se detuvo al momento de abrir o cerrar la válvula.

- Verificar las baterías se encuentren con carga, medir voltajes y corrientes.
- Verificar el mecanismo de los engranes.
- Realizar un reseteo o apagado de la válvula.

Contacto de soporte Técnico

Correo electrónico

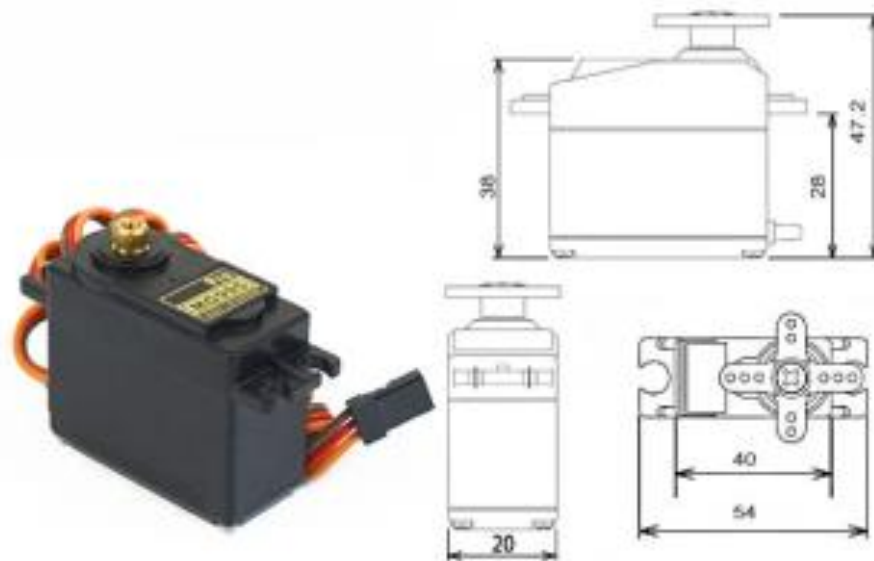
Juan Méndez juancamaneey@hotmail.com

Teléfono

Juan Méndez +593 996231497

ANEXO 6. DATASHEET

MG995 High Speed Metal Gear Dual Ball Bearing Servo



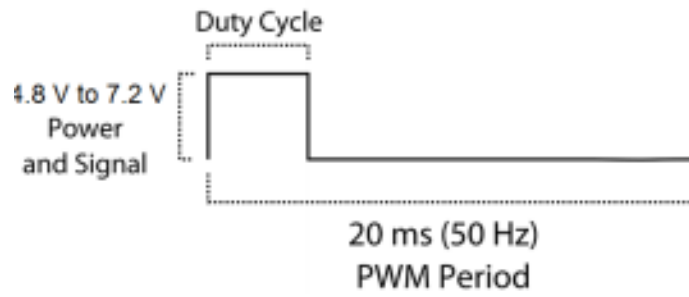
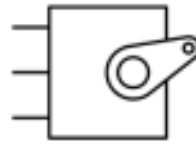
The unit comes complete with 30cm wire and 3 pin 'S' type female header connector that fits most receivers, including Futaba, JR, GWS, Cirrus, Blue Bird, Blue Arrow, Corona, Berg, Spektrum and Hitec.

This high-speed standard servo can rotate approximately 120 degrees (60 in each direction). You can use any servo code, hardware or library to control these servos, so it's great for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. The MG995 Metal Gear Servo also comes with a selection of arms and hardware to get you set up nice and fast!

Specifications

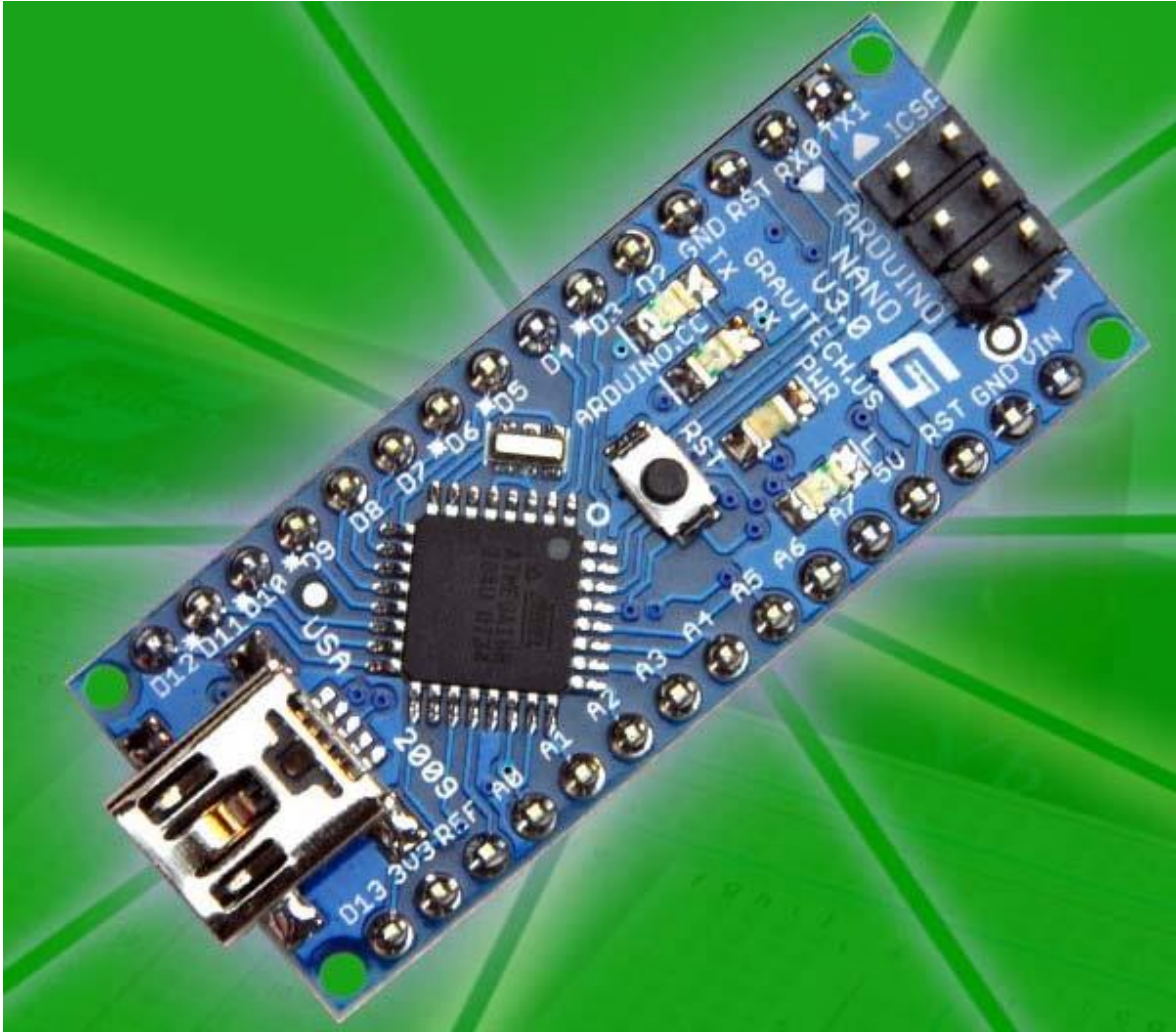
- Weight: 55 g
- Dimension: 40.7 x 19.7 x 42.9 mm approx.
- Stall torque: 8.5 kgf-cm (4.8 V), 10 kgf-cm (6 V)
- Operating speed: 0.2 s/60° (4.8 V), 0.16 s/60° (6 V)
- Operating voltage: 4.8 V a 7.2 V
- Dead band width: 5 μ s
- Stable and shock proof double ball bearing design
- Temperature range: 0 °C – 55 °C

PWM=Orange (⏏)
Vcc = Red (+)
Ground=Brown (-)

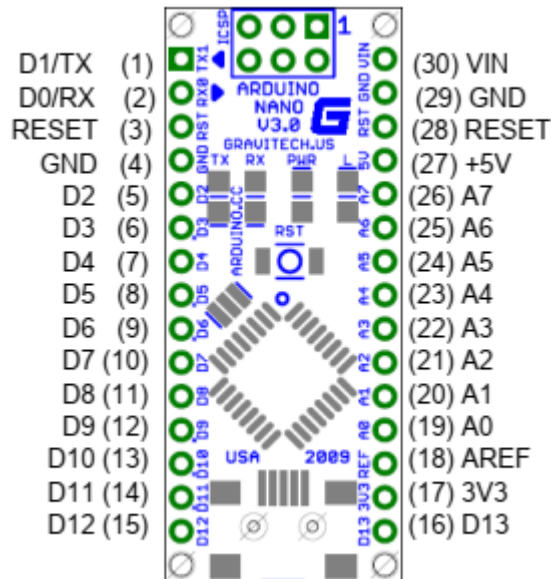


Arduino Nano (v3.0)

User Manual

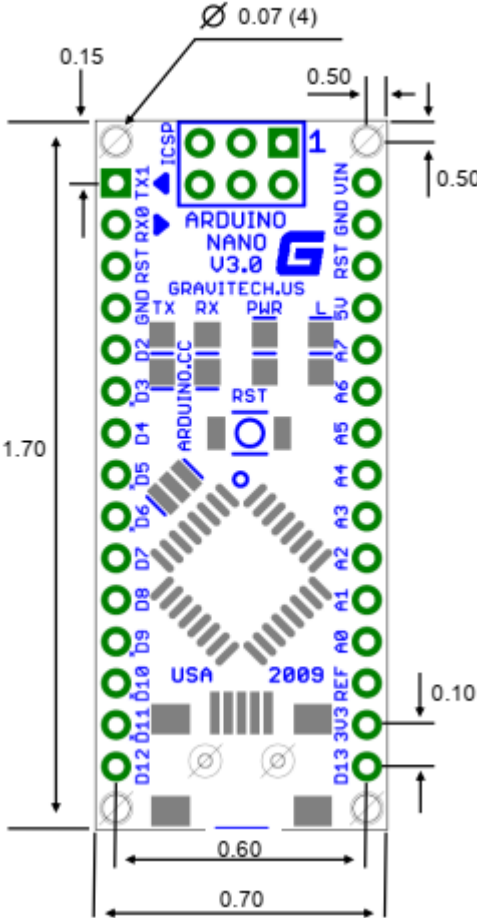


Arduino Nano Pin Layout

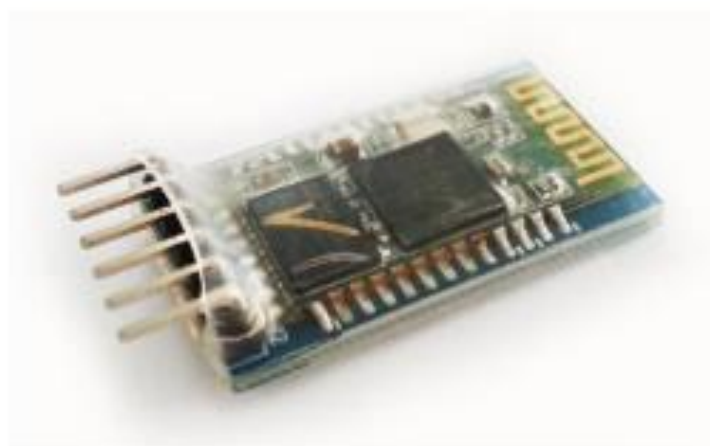


Pin No.	Name	Type	Description
1-2, 5-16	D0-D13	I/O	Digital input/output port 0 to 13
3, 28	RESET	Input	Reset (active low)
4, 29	GND	PWR	Supply ground
17	3V3	Output	+3.3V output (from FTDI)
18	AREF	Input	ADC reference
19-26	A0-A7	Input	Analog input channel 0 to 7
27	+5V	Output or Input	+5V output (from on-board regulator) or +5V (input from external power supply)
30	VIN	PWR	Supply voltage

Arduino Nano Mechanical Drawing



HC-05 Bluetooth Module



User's Manual V1.0

2. Pin Definition



Pin	Description	Function
VCC	+5V	Connect to +5V
GND	Ground	Connect to Ground
TXD	UART_TXD, Bluetooth serial signal sending PIN	Connect with the MCU's (Microcontroller and etc) RXD PIN.
RXD	UART_RXD, Bluetooth serial signal receiving PIN	Connect with the MCU's (Microcontroller and etc) TXD PIN.
KEY	Mode switch input	If it is input low level or connect to the air, the module is at paired or communication mode. If it's input high level, the module will enter to AT mode.

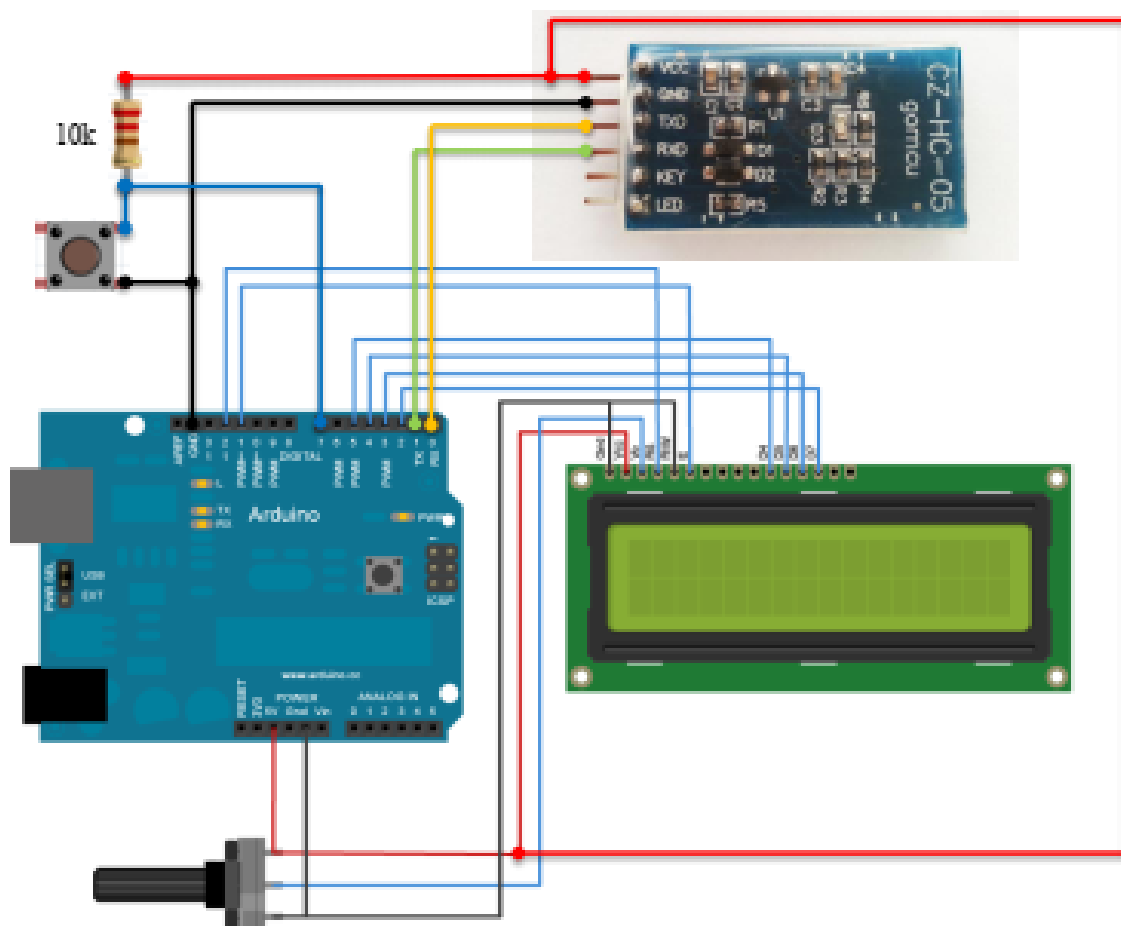
3. Sample Hardware Installation and Sample Source Code

Below is the example of interfacing between Arduino UNO and PC via HC-05 Bluetooth Module. In this example, the communication mode is used.

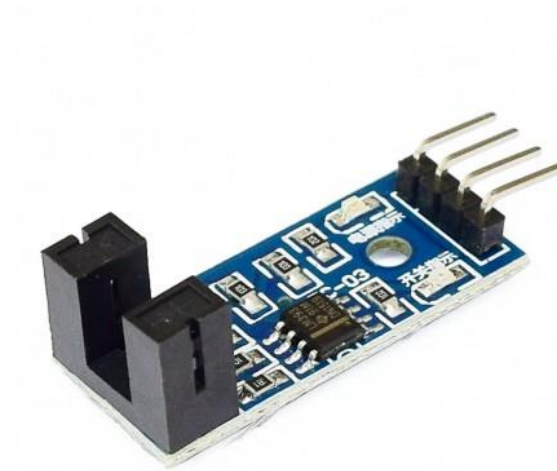
**Note: For AT mode (use to change the default setting or etc), please refer to this link [Modify The HC-05 Bluetooth Module Defaults: Using AT Commands](#).*

Diagram below shows the hardware connection between HC-05 Bluetooth Module and Arduino UNO. Besides Arduino, it may interface with any microcontroller such as PIC and etc.

- **VCC** → Arduino 5V
- **GND** → Arduino GND
- **TXD** → Arduino Pin RX
- **RXD** → Arduino Pin TX
- **KEY** → Connect to the air for communication mode



SENSOR DE VELOCIDAD HC-03



Conocer la posición o velocidad de un motor es muy importante en robótica, para lo cual existen diversas alternativas, siendo una de las más comunes el uso de encoders de tipo óptico. Los encoders en general son dispositivos que se encargan de convertir el movimiento angular o lineal en pulsos eléctricos que puedan ser interpretados por el controlador del sistema. Los encoders incrementales ópticos realizan la medición de movimiento con el uso de un haz de luz infrarrojo que se ve interrumpido por las ranuras de un disco acoplado al eje. La cantidad de ranuras por vuelta determinará la precisión del encoder.

Este sensor para encoder óptico utiliza el opto interruptor infrarrojo MOCH22A. El Dispositivo MOCH22A cuenta con dos partes: Un emisor IR y un receptor o sensor IR. Entre el emisor y receptor IR existe un espacio para las ranuras del disco. Los pulsos son digitalizados en un opamp LM393 entregando pulsos TTL que pueden ser interpretados por un microcontrolador como Arduino o Pic. Se recomienda utilizar **interrupciones** por flancos de subida/bajada para detectar los pulsos. En Arduino Uno los pines de interrupción por flanco son: 2 y 3. Se recomienda agregar un capacitor de 100nF entre la línea de salida D0 y tierra a modo de filtro pasa bajo y así evitar falsos disparos en la interrupción. Otra recomendación es alimentar el módulo con 3.3V.

Este modelo de encoder ha sido diseñado para trabajar con plataformas de robótica móvil: seguidores de línea, sumos, laberinto. También son utilizados en contadores de RPM (Revoluciones por minuto) en motores DC/AC o como sensor de final de carrera.

- Voltaje de Operación: 3.3V - 5V DC
- Salidas: Analógica y Digital TTL
- Sensor: MOCH22A
- Modelo Placa: FC-03 / FZ0888
- Tipo de emisor: Fotodiodo IR

- Tipo de detector: fototransistor
- Longitud de onda del emisor: 950 nm (infrarrojo)
- Peso: 8 gramos
- Dimensiones: 3.2*1.4*0.7 cm
- Ranura de 5mm
- Comparador Opamp: LM393
- Led indicador de alimentación
- Led indicador de pulso
- Salida TTL ON: Sensor bloqueado
- Salida TTL OFF: Sensor sin bloquear

DECLARACIÓN Y AUTORIZACIÓN

Yo, Juan Ignacio Méndez García, CI 1002438909 autor/a del trabajo de graduación:

Desarrollar una válvula de riego agrícola motorizada programable mediante bluetooth, autónoma y móvil, previo a la obtención del título de **Ingeniería Electrónica Digital y Telecomunicaciones** en la UNIVERSIDAD TECNOLÓGICA ISRAEL.

1.- Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de difundir el respectivo trabajo de graduación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la SENESCYT a tener una copia del referido trabajo de graduación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Quito, septiembre del 2018

Atentamente.

Juan Ignacio Méndez García.

C.I. 1002438909

