

DISEÑO E IMPLEMENTACIÓN DE VÁLVULA CONTROLADA MEDIANTE INTERFAZ WIFI

“DESIGN AND IMPLEMENTATION OF A VALVE CONTROLLED WITH WIFI INTERFACE”

Brayan Nedzib Sastoque Rangel¹, Samuel Raul Lopez Blanco², Andrés Escobar Díaz³

Resumen: En el presente documento se presenta un prototipo de válvula inteligente orientado a su uso en espacios domésticos. El modelo se desarrolló bajo 6 etapas: la modificación de su funcionamiento mecánico; el desarrollo electrónico y acondicionamiento eléctrico; el diseño de la programación necesaria para el funcionamiento y configuración; el establecimiento de un sistema de comunicación bidireccional a través de WiFi; el desarrollo de una página web, y todo lo requerido para su construcción física. Para el prototipo se procuró el uso de software y hardware libres. El proyecto busca generar un aporte al desarrollo en la domótica y el Internet de las cosas.

Palabras clave: Microcontrolador (ESP8266), comunicación WiFi, Válvula, interfaz, domótica, pagina web, Internet de las cosas, firebase.

Abstract: This paper presents a prototype of smart valve oriented to use in domestic spaces. The model was developed under 5 stages: the modification of its mechanical functioning; the electronic development and electrical conditioning; the design of the programming necessary for the operation and configuration; the establishment of a bidirectional communication system through WiFi, and the development of a web page. For the prototype, the use of free software and hardware was sought. The project seeks to generate a contribution to development in the home automation and Internet of things.

Keywords: Microcontroller (ESP8266), WiFi communication, Valve, interface, home automation, website, Internet of things, firebase.

¹ Estudiante Tecnología en Electrónica (Ciclos Propedéuticos). Universidad Distrital Francisco José de Caldas – Facultad Tecnológica. Email: bnsastoquer@correo.udistrital.edu.co

² Estudiante Tecnología en Electrónica (Ciclos Propedéuticos). Universidad Distrital Francisco José de Caldas – Facultad Tecnológica. Email: srlopezb@correo.udistrital.edu.co

³ Ingeniero En Electrónica, Magister en Administración, Magister en Ingeniería Electrónica, Grupo de Investigación en Orden y Caos ORCA. Universidad Distrital Francisco José de Caldas – Facultad Tecnológica. E-mail:

1. INTRODUCCIÓN

En la actualidad existe poco control sobre las variables en el hogar, puntualmente sobre el suministro principal de agua a pesar del creciente desarrollo en la tecnología domótica. Según el periódico EL TIEMPO para el 2016 cerca de 14 millones de personas tenían acceso a internet todos los días y al menos 9 millones que acceden una vez al mes, esto nos da 23 millones de usuarios [1] lo cual es casi la mitad de la población colombiana según el reloj poblacional del DANE [2]. Sin embargo, este acceso a internet se enfoca principalmente en el consumo de redes sociales [1]. En concordancia con CISCO la propuesta actual de la tecnología respecto a el Internet de las cosas, trata de usar las facilidades de conexión a la red para el control de sistemas físicos a través de la nube, así mismo estima una cantidad de 50 mil millones de dispositivos conectados a WiFi para el 2020 [3].

Desde otro punto de vista, el desarrollo de esta tecnología beneficia a la población en condición de discapacidad ya que las válvulas principales de agua contarán con un sistema de activación remota que facilita su manipulación mientras que las comunes se encuentran en una posición poco accesible para todas las personas, lo que afecta su calidad de vida. Según el portal 'Discapacidad Colombia' alrededor del 6% de la población colombiana tienen algún tipo de discapacidad y el 37% de estas personas requiere un cuidador o un tercero que los asista en algunas actividades de la vida diaria [4] por lo tanto, otro motivante para el avance tecnológico en la Domótica es el mejoramiento de la calidad de vida de las personas.

Este proyecto se implementa desde el Grupo de Investigación en Orden y Caos ORCA

2. MARCO TEÓRICO

2.1. MARCO CONCEPTUAL

2.1.1. Válvulas de Bola

Las válvulas de bola (Figura 1) ofrecen muy buena capacidad de cierre y son prácticas porque para abrir y cerrar la válvula es tan sencillo como girar la manivela 90°. Se pueden hacer de 'paso completo', lo que significa que

la apertura de la válvula es del mismo tamaño que el interior de las tuberías y esto resulta en una muy pequeña caída de presión. Otra característica principal, es la disminución del riesgo de fuga de la glándula sello, que resulta debido a que el eje de la válvula solo se tiene que girar 90°.

Cabe señalar, sin embargo, que esta válvula es para uso exclusivo en la posición totalmente abierta ó cerrada. Esta no es adecuada para su uso en una posición de apertura parcial para ningún propósito, tal como el control de caudal.

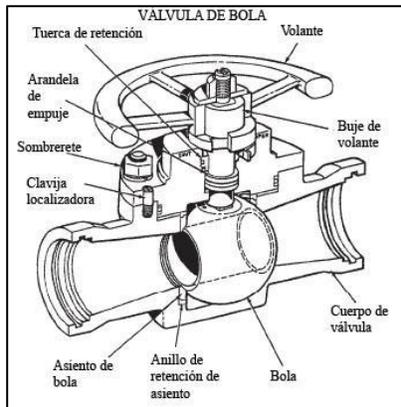


Figura 1. Válvula de Bola [5]

La válvula de bola hace uso de un anillo suave conformado en el asiento de la válvula. Si la válvula se utiliza en posición parcialmente abierta, la presión se aplica a sólo una parte del asiento de la válvula, lo cual puede causar que el asiento de la válvula se deforme. Si el asiento de la válvula se deforma, sus propiedades de sellado se vulneran y esta fugará como consecuencia de ello. [6]

2.1.2. Redes de área amplia

Una red de área amplia (WAN), es una red de comunicación que abarca una gran área geográfica. Contiene un conjunto de máquinas diseñado para programas (es decir, aplicaciones) de usuario.

Los usuarios están conectados por una subred de comunicación, o simplemente subred, para abreviar. Los clientes son quienes poseen al host (es decir, las computadoras personales de los usuarios), mientras que, por lo general, las compañías telefónicas o los proveedores de servicios de Internet poseen y operan la subred de comunicación. La función de una subred es llevar mensajes de un host a otro, como lo hace el sistema telefónico con las palabras del que habla al que escucha [7]. La red de Área amplia es en su aspecto más simple la Internet Abierta, todas las páginas y lugares virtuales que abarca una conexión a la red mundial.

2.1.3 Sistemas Embebidos

Un sistema embebido es un sistema basado en microprocesador que está construido para controlar una función o rango de funciones y no es diseñado para ser programado por el usuario final de la misma manera que una PC. Sí, un usuario puede tomar decisiones relacionadas con la funcionalidad, pero no puede cambiar la funcionalidad del sistema añadiendo / reemplazando software. [8] Es poner en un sistema hardware y software de computadora para una función particular.

2.1.4 NODE MCU (ESP8266) [11]

El sistema de Hardware Libre NODE MCU (figura 2) es una board de desarrollo basada en el microcontrolador ESP8266-12e que cuenta con salidas análogas y una entrada análoga. Dicho dispositivo se programa usando lenguaje LUA aunque también existe un entorno de programación con las librerías necesarias para su programación en lenguaje C++. Node Mcu integra un controlador USB-Serial el cual se usa para programación y comunicación, también cuenta con un regulador de tensión el cual facilita el uso del microcontrolador al ampliar su voltaje de operación (3.3V a 20V) [9] la característica más importante de este microcontrolador es su capacidad de conexión a internet a través de WiFi así como la posibilidad de configurar su funcionamiento tanto como Cliente o como Servidor Web.

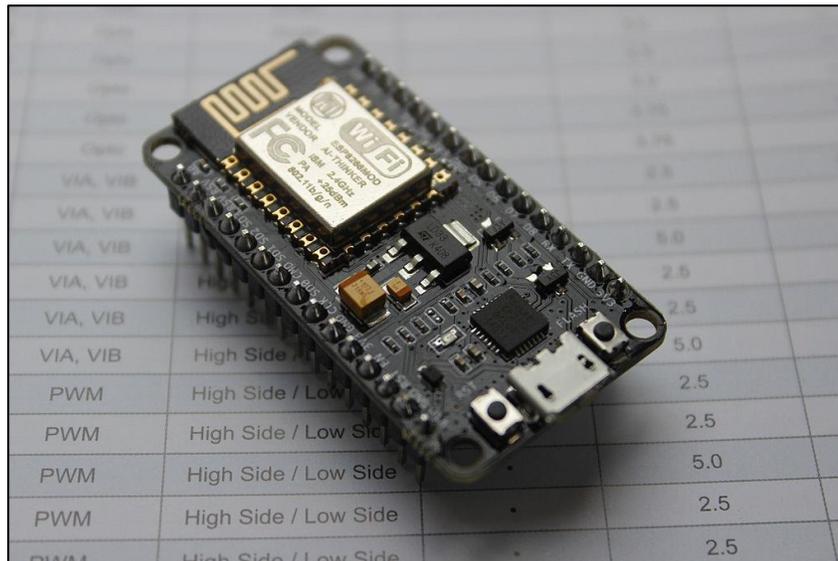


Figura 2. NODE MCU[10]

2.1.5 Firebase Realtime Database

Firestore Realtime Database es una base de datos alojada en la nube. Los datos se almacenan en formato JSON y se sincronizan en tiempo real con cada cliente conectado. Cuando se compilan apps multiplataforma con los SDK de iOS, Android y JavaScript, todos los clientes comparten una instancia de Realtime Database y reciben actualizaciones de forma automática con los datos más recientes.

En lugar de solicitudes típicas de HTTP, Firestore Realtime Database usa la sincronización de datos (cada vez que los datos cambian, los dispositivos conectados reciben esa actualización en milisegundos). Proporciona experiencias colaborativas y envolventes sin pensar en el código de red. [12]

3. CONSTRUCCIÓN DEL PROTOTIPO

3.1. Diagrama de bloques

En la figura 3 presentamos el diagrama de bloques desarrollado para el prototipo (figura 4) realizado.

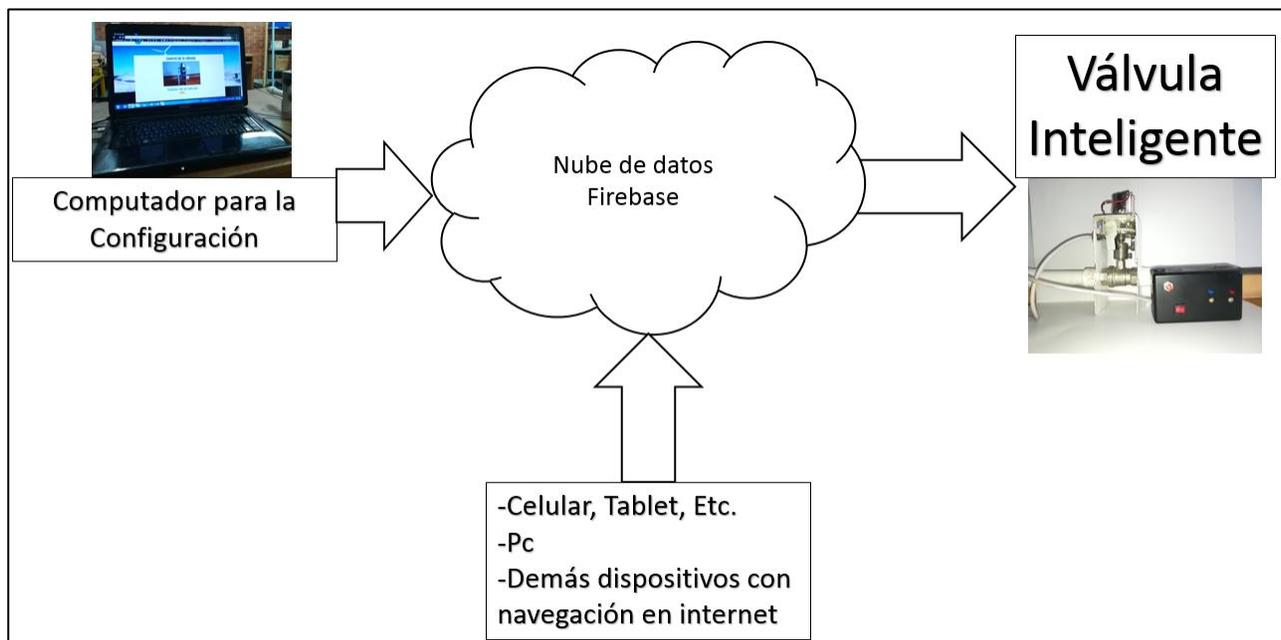


Figura 3. Diagrama de bloques

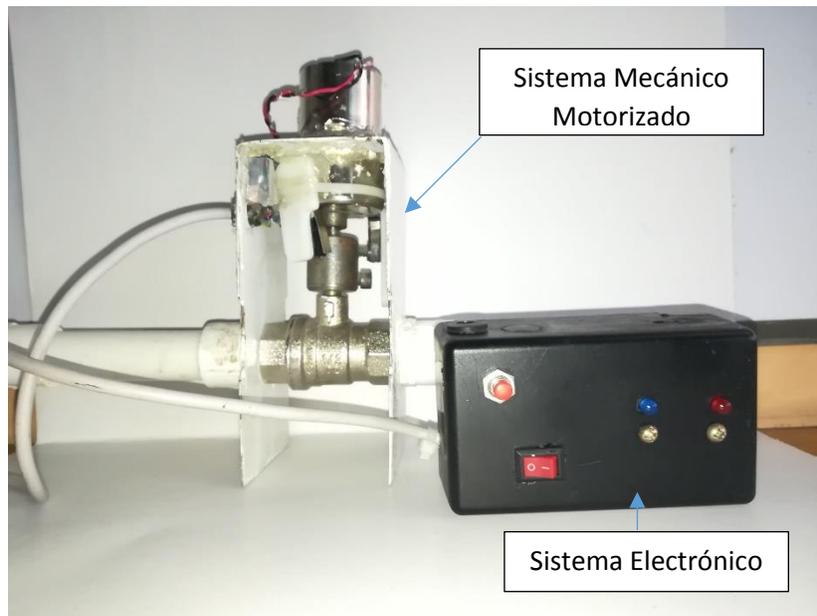


Figura 4. Prototipo

3.2 Funcionamiento Mecánico

En la figura 5 se muestra el acople motorizado aplicado para el funcionamiento mecánico de una válvula de bola de ½ pulgada. En primer lugar, se retira la mariposa de la válvula, luego con una junta metálica se realiza el empalme entre el eje mecánico de un sistema eléctrico con el de la válvula, para este proyecto se cuenta con un motor de 12 voltios el cual tiene motoreductor, lo que aumenta el torque para abrir y cerrar la válvula con facilidad.

Para asegurar el movimiento en la válvula se mantiene detenido el motor con un soporte cuadrado sostenido por la estructura de la red de agua.

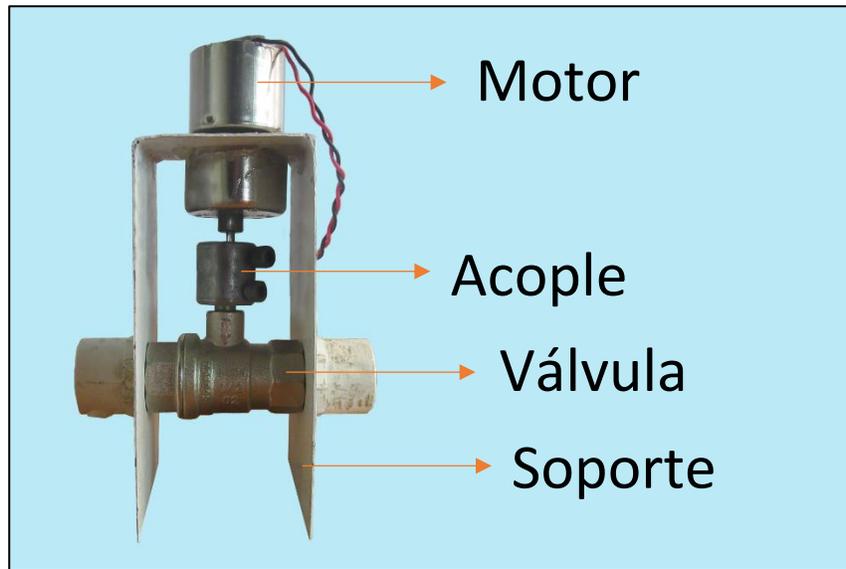


Figura 5. Acople Motorizado

3.3 Desarrollo Electrónico y Acondicionamiento Eléctrico

3.3.1 Desarrollo Electrónico

En primer lugar, tenemos que integrar al sistema tanto el microcontrolador como un inversor de giro para el motor (puente H), pensando en que el prototipo sea compacto se usa el integrado L293D con voltaje de operación de 4.5V hasta 36V, el L293D cuenta con dos inversores de giro, soporta una corriente máxima de 1200mA y el motor escogido requiere cerca de 250mA para su funcionamiento a 12V. Su conexión se realiza como se muestra en la Figura 6 en la cual también está la conexión de 2 leds que sirven de testigos para el movimiento de la válvula.

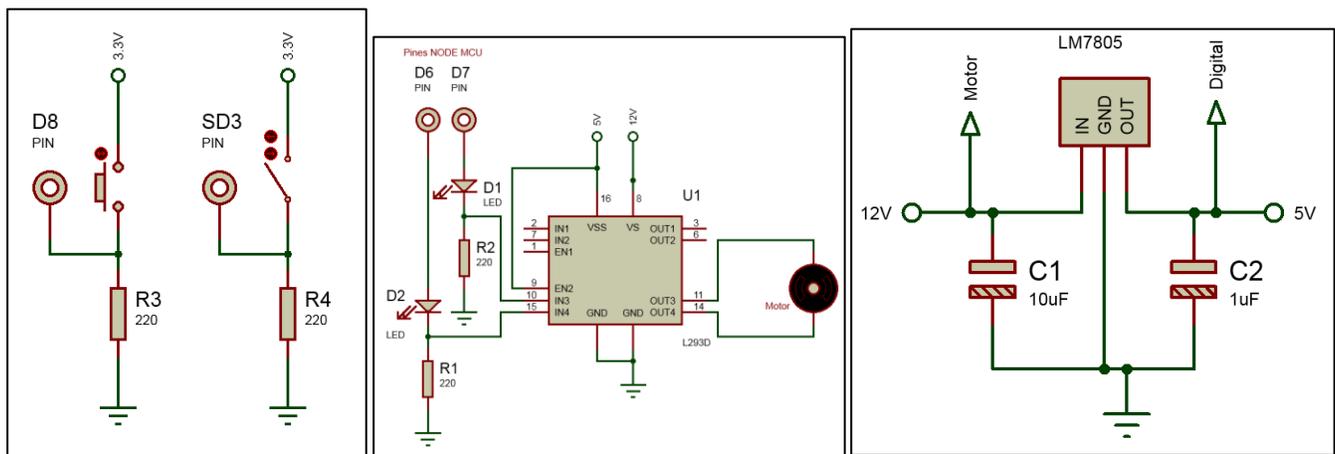


Figura 6. Izquierda conexión de botón e interruptor, Centro circuito para L293D, Derecha regulador de tensión.

Luego, para hacer uso de un modo de configuración se añade al circuito un interruptor y también un botón para la apertura manual de la válvula (Figura 6), así como dos finales de carrera para identificar cuando el motor cierra o abre.

Por último, la integración de un motor implica una generación de ruido cada vez que éste se mueva, para solucionarlo se ubica un condensador electrolítico entre la alimentación de 12V y Tierra.

3.3.2 Acondicionamiento Eléctrico

Para la conexión a la red eléctrica se ocupa un adaptador DC de 12V de 1.25A, sin embargo, para manejar niveles seguros de voltaje el sistema digital y el integrado se alimentan con 5V mientras el motor con 12V esto se hace posible usando un integrado regulador de tensión, para este caso el LM7805, en la Figura 6 se observa su diagrama de conexión.

3.4 Software, Programación, Conexión WiFi

Cabe aclarar que el lenguaje escogido para la programación del microcontrolador fue C++ y el de la interfaz de configuración Python3.

3.4.1 Configuración del Microcontrolador

Inicialmente se tiene el problema de una memoria volátil y la necesidad de conectarse a la red cada vez que se reinicie el sistema, la solución a esto la tiene el microcontrolador el cual cuenta con una memoria EEPROM de lectura y escritura. De igual forma, el sistema tiene que ser amigable, por lo cual se programa en Python una interfaz gráfica de usuario para facilitar esta configuración.

Inicialmente se alterna el interruptor que provoca que al conectar con cable de datos al computador el prototipo entre en estado de configuración, luego mediante comunicación serial RS232 se envían los datos necesarios tanto para asegurar la conexión entre la interfaz y el dispositivo como para la configuración del mismo.

En el diagrama de flujo de la Figura 7 se imprime la escritura de la memoria EEPROM, para este fin se usó lenguaje C++ y la librería EEPROM.h

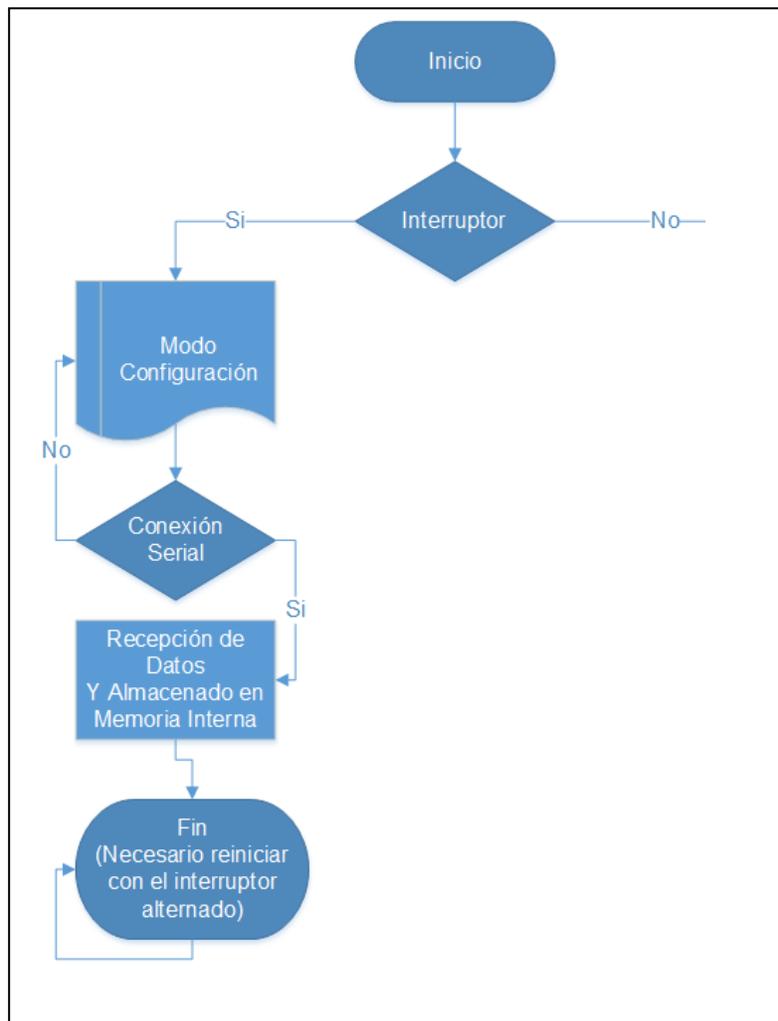


Figura 7. Diagrama Modo Configuración

Cuando el prototipo recibe una 'c' desde el pc es porque se ha asegurado una conexión confiable entre los dos sistemas y luego comienza a guardar en cada espacio de memoria EEPROM lo que llega al serial, el formato escogido para los datos fue el siguiente:

NOMBREDERED[CONTRASEÑA]

Y cuando llega el carácter ']' reconoce esto como el fin de los datos y lo anuncia enviando mensajes al computador.

Desde el computador la interfaz diseñada en Python (Figura 8) usando la librería nativa Tkinter para el diseño, tiene únicamente 2 botones para ser más sencilla de usar, uno para el envío de datos y el otro para salir de la interfaz.

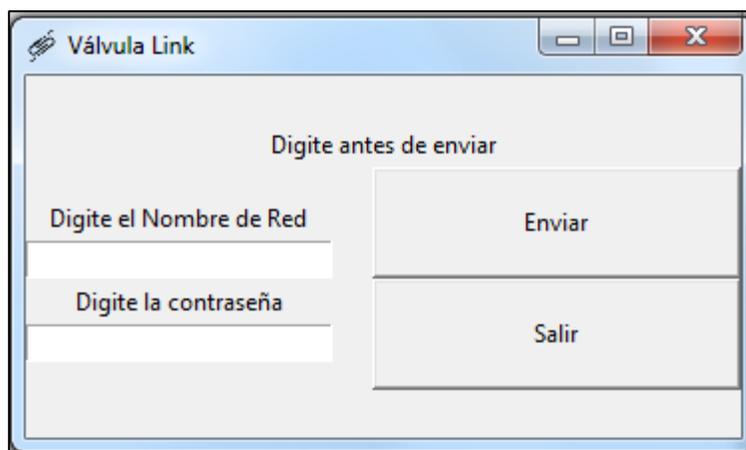


Figura 8. Interfaz de Usuario

Una vez que el usuario haya digitado el nombre de red y la contraseña de la conexión WiFi que usara el prototipo oprime el botón Enviar, al momento de hacer esto se muestra una alerta de escritorio (Figura 9) en la que se informa que él envió de datos puede tardar hasta 1 minuto, pero, esto solo es por prevención ya que el microcontrolador escribe la memoria física a cerca de 1 dato cada 13 milisegundos, al finalizar, se muestra otra alerta de escritorio (Figura 9) en la cual se indica que fue un proceso exitoso.

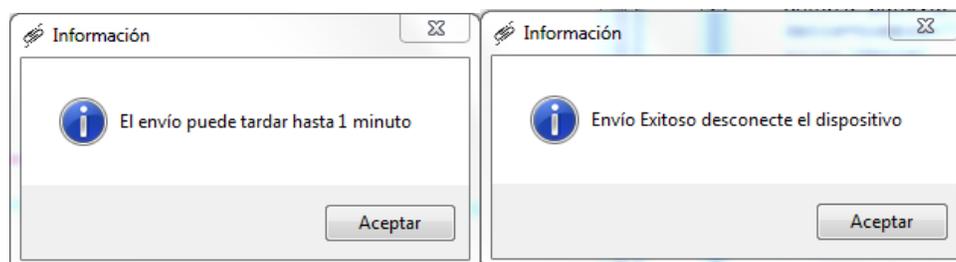


Figura 9. Alertas de Escritorio

En la programación el primer método importante es el de escaneo de los puertos seriales, El dispositivo en estado de configuración mantiene enviando una 'c', la interfaz comienza a leer cada puerto COM desde el 0 hasta el 255 y guarda en una variable global el valor del puerto desde el cual recibió una 'c', es decir, el puerto de nuestro dispositivo. En el diagrama de la figura 10 se puede observar el funcionamiento de la interfaz gráfica.

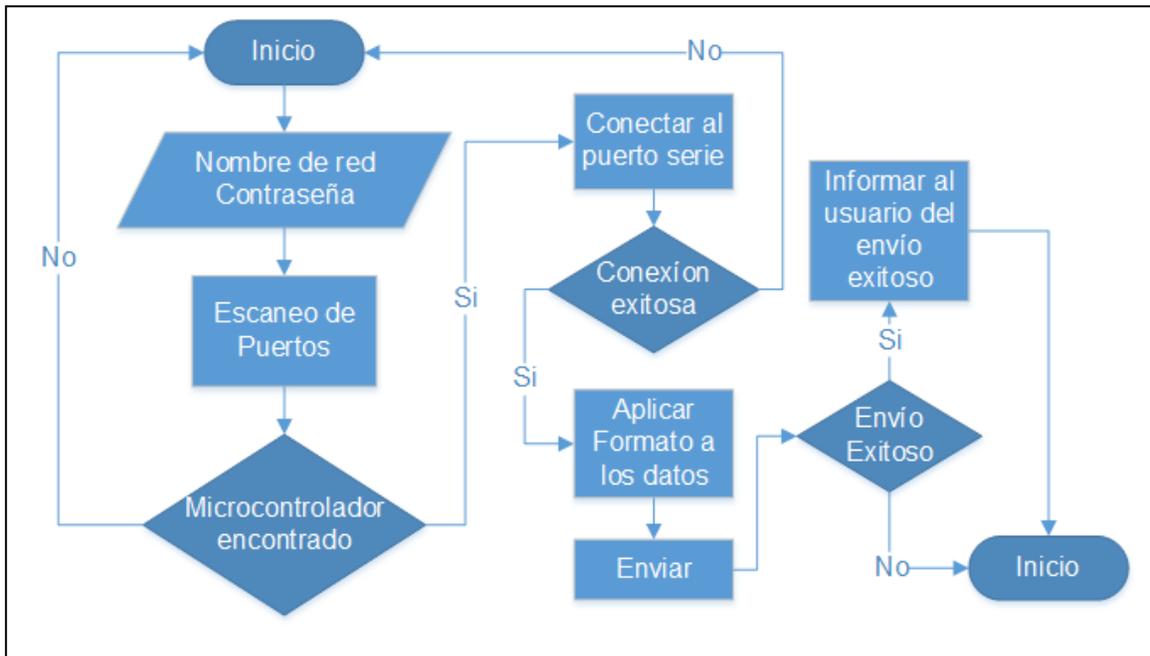


Figura 10. Diagrama Interfaz Gráfica

A continuación se establece una conexión al puerto donde encontró el prototipo enviando una 'c', luego se envían los datos a través de comunicación serial bajo el formato establecido y cuando el microcontrolador envía los mensajes al haber terminado de guardar los datos, la interfaz termina su función.

3.4.2 Conexión a la red WiFi

Luego del envío y guardado de datos, en la primera etapa de conexión se recupera el nombre de red y contraseña, el código (Figura 11) no es más que un ciclo donde se lee cada espacio de memoria identificando los caracteres '[' y ']' para separar el 'nombre de red' de la 'contraseña', luego de terminar se cambia el valor de una variable de control y esto provoca el fin del ciclo.

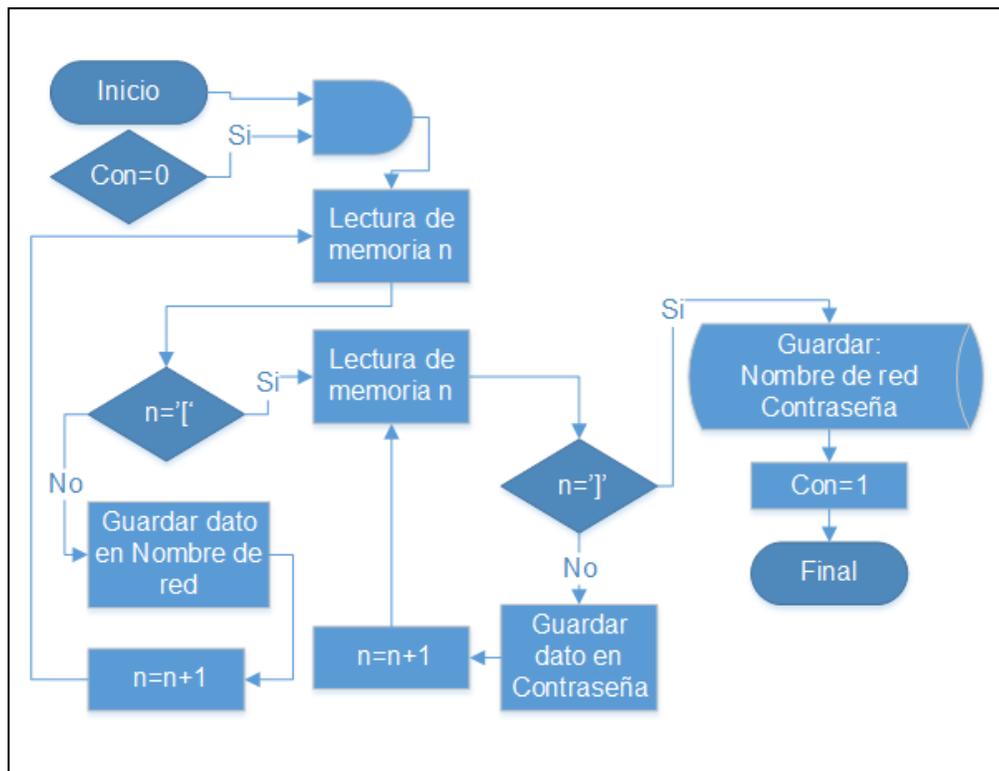


Figura 11. Lectura EEPROM

Para el uso del microcontrolador (figura 12) se ocupa la librería ESP8266WiFi.h la cual tiene los métodos necesarios para conectarse a la red, este método intenta hacer la conexión con los datos leídos y guardados, si la conexión es fallida intenta nuevamente hasta que sea posible establecerla.

3.4.3 Configuración en Red de Firebase

En Firebase se guardarán las variables importantes para el control on/off de la válvula. Este también nos permite asegurar una comunicación bidireccional entre la página web que controlara la válvula y el microcontrolador. Firebase es una interfaz intuitiva para los desarrolladores en donde las variables y todo lo que se quiera almacenar allí se guarda en formato JSON bajo el aspecto visual de listas.

Para el control de la válvula se escoge usar dos variables, esto, ya que la variable de control nos ayuda a la sincronización de la válvula con el estado presente en la página web, es decir, tenemos dos variables que actúan como binarias una numérica que varía entre 0 y 1 donde su valor solo cambia cuando se ha efectuado el cambio de estado (abierta o cerrada) de la válvula, la otra booleana que es la que alterna el usuario. Esta interacción se puede ver en la tabla de estados de la figura 12, si el sistema está sin alimentación y el usuario alterna el estado de la

válvula desde la red, cuando se vuelva a prender, esta lee el estado y se ajusta a lo indicado por el usuario.

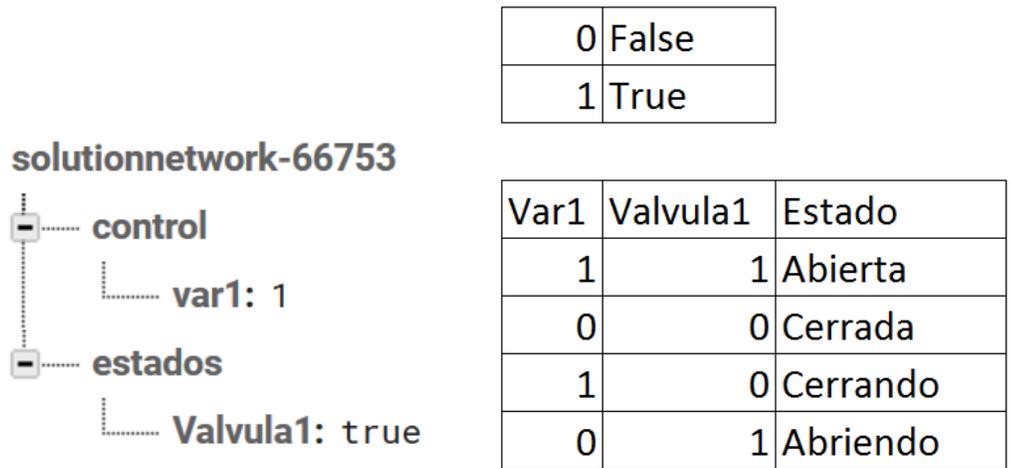


Figura 12. Variables y Tabla de estados

3.4.4 Conexión con Firebase y Control Web

En el ciclo principal se encuentran dos fases, la primera el control web a través de Firebase (figura 15). Firebase genera dos constantes, una url de donde están nuestras variables y una llave para autenticar las aplicaciones o dispositivos que generaran cambios en las variables, en el código del microcontrolador se guardan estos valores y se procede a iniciar la conexión con Firebase, la lógica utilizada es la de la figura 13. La segunda fase hace referencia al control manual.

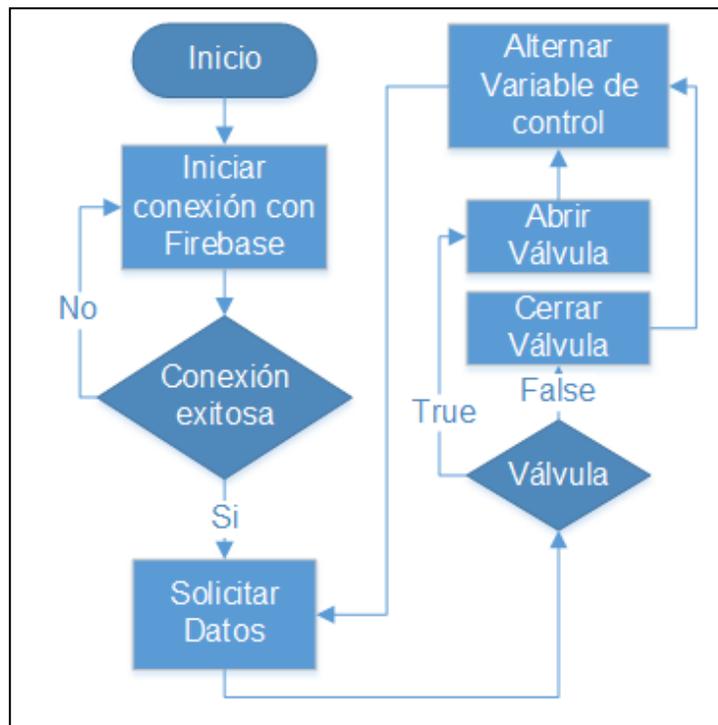


Figura 13. Conexión y control web con Firebase

Luego de hacer la lectura se emplea una salida digital para activar el funcionamiento del motor hasta que éste accione un sensor de final de carrera; adicionalmente se colocan las líneas de código que modifican el estado de la variable de control en Firebase. Cabe aclarar que cuando falla esta modificación de estado, el código lo intenta las veces que sean necesarias hasta que se efectúe.

3.4.5 Control Manual de Válvula

La segunda parte importante del ciclo principal es el control manual de la válvula, para esto se utiliza una entrada digital con el fin de reconocer cuando se oprime un botón, luego, se ejecuta el mismo código para apertura web con la variante de que al finalizar el movimiento cambia el estado de las dos variables en Firebase. (figura 14)

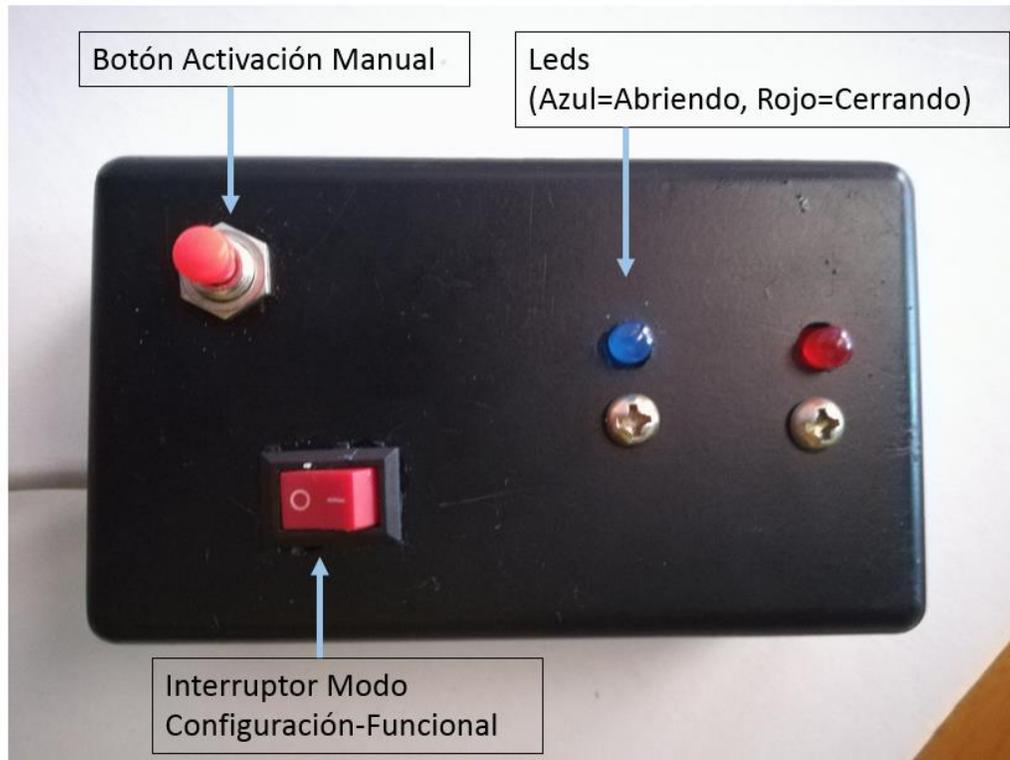


Figura 14. Interfaz física para activación y control manual

3.4.6 Diagrama de Flujo General

Ahora podemos establecer un diagrama de flujo que explique de forma general el proceso creado para la válvula en la cual los cuadros verdes son donde cambia de estado la válvula; los amarillos son donde se interactúa con Firebase y el naranja indica el ciclo principal. (Figura 15)

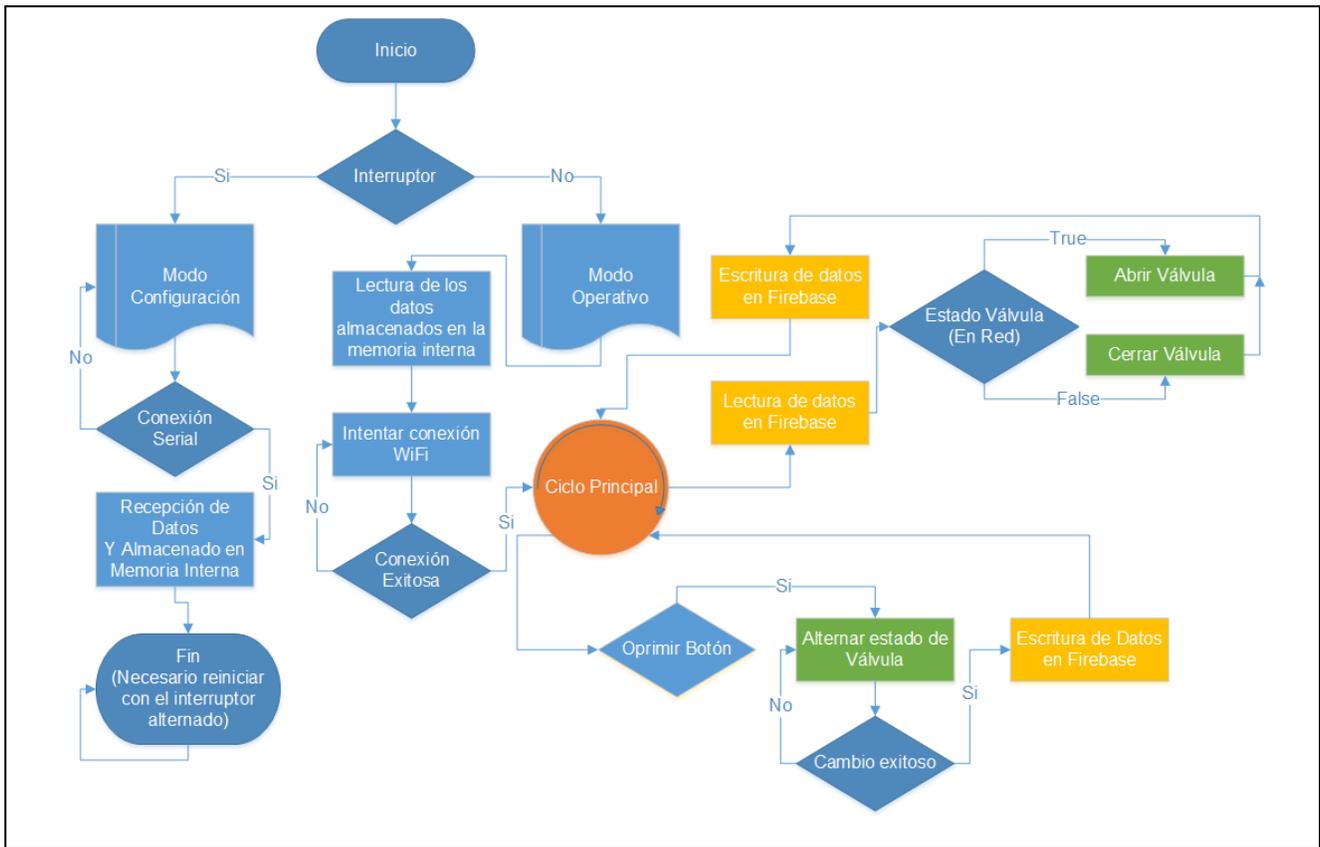


Figura 15. Diagrama de flujo Válvula

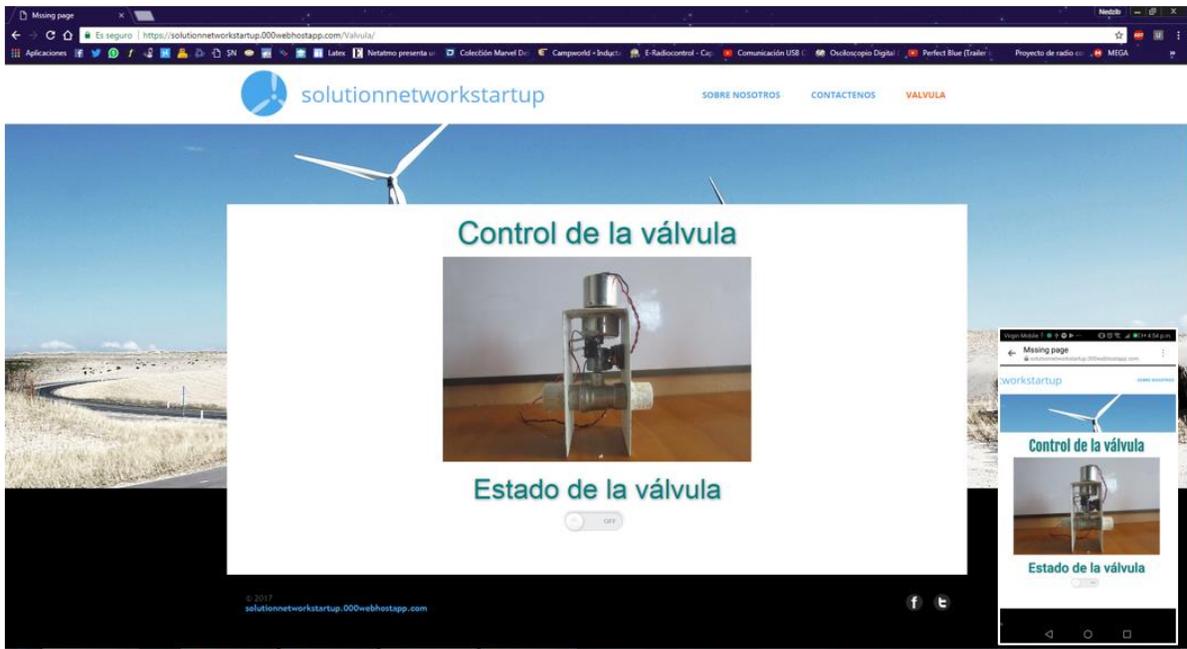
3.5 Diseño de Página web

Para lograr un control a través de un dominio web, estando en Firebase se da clic a la opción de añadir Firebase a tu aplicación web. Esto genera un código en formato JavaScript que se puede incrustar a una página web que soporte este lenguaje.

Para este proyecto, se ocupó un host y un dominio gratuitos de la página www.000webhost.com. El dominio que controlara la válvula es:

<https://solutionnetworkstartup.000webhostapp.com/Valvula/> (imagen 1)

La interfaz en la página es solamente un interruptor virtual desde el cual el usuario alterna el estado de la válvula.



Página desde PC

Página desde Celular

Imagen 1. Página Web

3.6 Construcción del Prototipo

3.6.1 Circuito Impreso

El diseño se realizó en el programa Proteus 8.4 y su impresión fue en el Laboratorio de Circuitos Impresos de la universidad Distrital Francisco José de Caldas – Facultad Tecnológica. El diseño creado es el que se muestra en la figura 16 y 16.1.

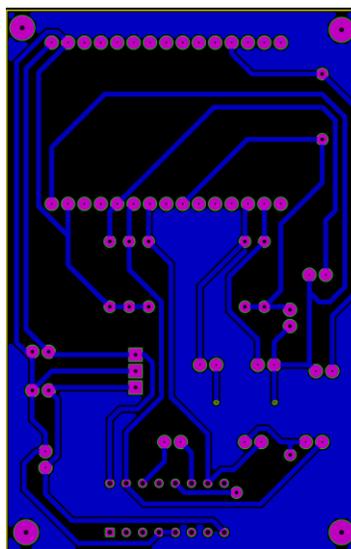


Figura 16. Circuito Impreso

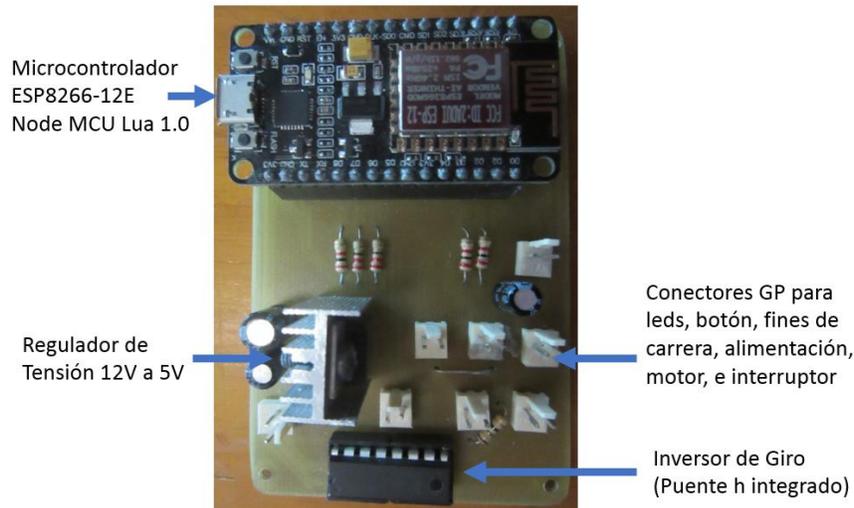


Figura 16.1 Circuito Impreso

3.6.2 Acondicionamiento del espacio

En primer lugar y con el objetivo de hacer un dispositivo compacto, seguro y fácil de usar se opta por una estructura (figura 4) compuesta por dos partes conectadas por los cables necesarios, estas partes corresponden al sistema mecánico motorizado y al sistema electrónico.

4. PRUEBAS Y RESULTADOS

Inicialmente se realiza una caracterización básica del dispositivo, luego se realizan pruebas de funcionamiento preliminares y a tiempo extendido identificando las veces que se reinicia el dispositivo a causa del Watchdog Timer y el tiempo promedio que demora para volver a su estado funcional.

4.1 Caracterización del Prototipo

Aquí se consignan aspectos físicos y de funcionamiento principales del prototipo (tabla 1) al vacío (sin flujo de agua) para una conexión a internet de 10mb/s.

PARÁMETRO (Visto desde el frente)	Válvula Inteligente
Altura	16.5 Cm aprox.
Ancho	6 Cm aprox.
Profundidad	7.3 Cm aprox.
Peso del prototipo (con terminales de agua)	635 gr

Voltaje de Operación	12V
Corriente recomendada	1.25A
Corriente del motor	250mA
Tamaño de Válvula	Media Pulgada
Tiempo de apertura	1.48 Seg aprox.
Tiempo de cierre	2.19 Seg aprox.
Tiempo de respuesta (Apertura por Red)	1.18 Seg aprox.

Tabla 1. Caracterización del Prototipo

4.2 Pruebas preliminares

En las primeras pruebas realizadas a nuestro prototipo se cumple con la ficha de caracterización, esto, antes de realizar el diseño de pcs para facilitar su manipulación en la protoboard.

4.3 Pruebas de tiempo extendido

En estas pruebas de largo tiempo se diseña la versión compacta de nuestro prototipo donde todo está diseñado según lo planificado, se consigue que el dispositivo dure conectado 1 mes de forma intermitente sin presentar ninguna falla con la comunicación, ejecutando las instrucciones remotas y manuales de forma adecuada.

4.4 Prototipo final

Esta última versión separa lo electrónico de lo mecánico dándole mayor orden y estética a nuestro producto logrando dar por terminado el proyecto, esta versión lleva 1 semana conectada donde no muestra fallos y hace más amigable su conectividad con el usuario.

4.5 Resultados adicionales

Un prototipo funcional, controlable las 24 horas de fácil configuración mediante una interfaz en Python y con fácil replicabilidad.

Una prueba adicional bajo la presión normal del agua en un ambiente domestico mostro que la válvula es capaz de funcionar en estas condiciones, sin embargo, hay que tener en cuenta que en todas las unidades domesticas la presión puede ser distinta.

Perspectivas y sugerencias

5. CONCLUSIONES

- El desarrollo del proyecto fue un éxito gracias al grupo de investigación en orden y caos ORCA.
- Se cumplió a cabalidad con los objetivos y cronograma propuesto.
- La comunicación utilizando firebase como interlocutor entre ambos medios hace más fácil la implementación y desarrollo del mismo.
- En el desarrollo del programa para el microcontrolador se evidencia un error en todo el guardado de datos en la EEPROM para el caso particular de que el nombre de red contenga corchetes cuadrados “[]”.
- Este proyecto teóricamente se puede desarrollar sin usar un regulador de voltaje (para 5V), sin embargo, se usa para que el sistema en el microcontrolador no se caliente cuando se mantenga por horas y días conectado.
- La presencia de una red (así no tenga acceso a internet) es necesaria para la manipulación manual de la válvula.
- A causa de la fuerza necesaria para la activación de la válvula, el funcionamiento mecánico fue desarrollado con un motor dc y no con un servo motor como se tenía planificado.

REFERENCIAS

[1]<http://www.eltiempo.com/archivo/documento/CMS-16649959>

[2]<http://www.dane.gov.co/reloj/>

[3]https://www.cisco.com/c/dam/global/es_mx/solutions/executive/assets/pdf/internet-of-things-iot-ibsg.pdf

[4]<http://discapacidadcolombia.com/index.php/discapacidad-y-sexualidad>

[6]<http://www.tlv.com/global/LA/steam-theory/types-of-valves.html>

[7]<https://julioestrepo.files.wordpress.com/2010/08/redes-de-computadoras-tanenbaum-4ta-edicion-espanol.pdf>

[8]<http://read.pudn.com/downloads158/ebook/707037/Embedded%20Systems%20Design%20-%20ed%20-%200750655461.pdf>

[9]https://github.com/nodemcu/nodemcu-devkit-v1.0/blob/master/NODEMCU_DEVKIT_V1.0.PDF

[10]https://raw.githubusercontent.com/nodemcu/nodemcu-devkit-v1.0/master/Documents/NodeMCU_DEVKIT_1.0.jpg

[11]http://nodemcu.com/index_en.html

[12]<https://firebase.google.com/docs/database/>

Imágenes

[5]<http://2.bp.blogspot.com/-LwuS5UZvd1k/VST7yILfsFQI/AAAAAAAAAP4/8KKgMt-19zU/s1600/valvula-de-bola.jpg>