

Universidad Nacional de Ingeniería
Facultad de Ciencias
Escuela Profesional de Ingeniería Física



Proyecto de investigación

“Desarrollo de un sistema para el control y monitoreo de una bomba de jeringa”

Presentan:

Polti Castillo Oscar Bruno

Profesores:

German Comina Bellido

Oswaldo Rojas Estela

Arnold Anthony Huaman Aguirre

Lima-2022

Resumen

En el presente proyecto se realizó un prototipo de bomba de infusión controlado de manera remota a través de un aplicativo web tomando de soporte la base de datos de Google Firebase. Asimismo, se ha trabajado con un sistema mecánico adecuado que permita manejar y calibrar su funcionamiento.

Palabras Clave: Jeringa, infusión, Firebase.

Abstrac

In this project, a prototype of a remotely controlled infusion pump was made through a web application using the Google Firebase database. Likewise, an adequate mechanical system was used to allow handling and calibration its operation.

Keywords: Syringe, infusion, Firebase.

ÍNDICE

1. Objetivos	4
1.1 Objetivos generales	4
1.2 Objetivos específicos	4
2. Importancia del proyecto	4
2.1 Antecedentes	4
2.2 Problema identificado	5
2.3 Hipótesis de trabajo	5
3. Marco teórico	5
4. Desarrollo del proyecto	11
4.1 Componentes y presupuesto estimado	11
3.1 Sistema mecánico	12
3.1.1 Jeringa	12
3.1.2 Carcasa para la bomba de jeringa	14
3.2 Motor DC, CD-ROM	15
3.2.1 Tornillo de avance	16
3.3 Esquema del proyecto	16
3.4 Procedimiento y conexiones	16
3.5 Drive controlador Puente H.	18
4 Aplicativo Web y Base de datos	19
5 Diagrama de Gantt	25
6 Resultados	25
7 Discusión de resultados.	26
8 Conclusiones	28
Referencias	28

1. Objetivos

1.1 Objetivos generales

Diseñar, implementar y calibrar un sistema remoto de control y monitoreo de una bomba de infusión.

1.2 Objetivos específicos

- Programación del aplicativo web para el control remoto de la bomba de infusión.
- Enlazar el aplicativo web con la base de datos de Google Firebase y con el microprocesador.
- Realizar las conexiones y programar un motor CD-ROM para el desplazamiento del sistema en una dimensión.
- Construir el prototipo apoyándonos de algunas piezas hechas en impresora 3D.
- Realización de las pruebas con parámetros definidos.

2. Importancia del proyecto

Debido al contexto en que nos encontramos, como es el de una pandemia, así como la poca accesibilidad a algunas zonas rurales; hacen que la aplicación de la telemedicina se haga una necesidad más frecuente. Es por ello, que este proyecto pretende ayudar a entregar fluidos como nutrientes y medicamentos al cuerpo de un paciente en un ambiente controlado.

2.1 Antecedentes

La bomba de infusión de jeringa es una herramienta de precisión en la administración de medicamentos basados en infusión a pacientes de cuidados intensivos. Permite tener descargas controladas de medicamentos. También controla el período de infusión de medicamentos según sea necesario. Un operador puede usar la bomba manualmente o usando nuestro enlace desarrollado o a través de una aplicación móvil dedicada que aumenta su accesibilidad [1].

Los avances en tecnologías inalámbricas e Internet de las cosas (IoT) facilitan el uso de dispositivos médicos de forma inalámbrica para que se pueda acceder a ellos a larga distancia para telemonitoreo y telemedicina. Entonces, para mantener el registro de los pacientes y el estado de su jeringa, hemos introducido IoT (Internet de las cosas) en este equipo médico. El IoT ayuda al Doctor a realizar un seguimiento del estado de la jeringa [2]. El IoT proporciona acceso remoto a la bomba de jeringa desde la cabina del médico, lo que ahorra mucho tiempo y esfuerzo humano. Esto reduce la carga de trabajo del personal del hospital en términos de mano de obra requerida y reducción en la intensidad requerida de la tarea. Esto ahorra recursos y tiempo simultáneamente [3].

En [4] diseñaron una bomba de jeringa modificada y la denominaron bomba de silicio de alto rendimiento para el sistema de administración de fármacos. [7] En la se observó que la capacidad de respuesta de la jeringa aumentaba al disminuir el volumen de la jeringa. En [5] trabajó en el control de la bomba de jeringa de acción simple utilizando el microcontrolador

8085. En 8085, la depuración y detección de errores del microcontrolador era compleja e involucraba un sistema de codificación hexadecimal.

Los mecanismos de accionamiento de la bomba de jeringa desempeñan un papel importante en el control de la bomba de jeringa. En la Ref. [6] se observó que el control lineal del tornillo de avance era posible con dispositivos de baja velocidad con movimientos precisos. En el mecanismo de husillo, el eje del motor estaba conectado a un tornillo roscado y, mientras gira, mueve la tuerca hacia adelante o hacia atrás y también el pistón conectado con ella. Por lo tanto, el movimiento del eje gobierna la entrega de fluido de manera controlada. El pistón generó una presión hidrostática, que entregó el fluido a la cámara.

2.2 Problema identificado

Debido al uso más frecuente de la telemedicina, es necesario un instrumento que facilite la administración de medicinas o nutrientes a los pacientes de una manera eficiente, en las dosis correctas y con la mayor precisión, de una manera remota.

2.3 Hipótesis de trabajo

Realizaremos una automatización y control del funcionamiento de una bomba de jeringa utilizando IoT así como un motor dentro de una carcasa donde la jeringa haga su proceso de forma unidimensional.

3. Marco teórico

SISTEMAS EMBEBIDOS:

Un sistema embebido es un sistema de computación basado en un microprocesador o un microcontrolador diseñado para realizar una o algunas pocas funciones dedicadas. Fueron fabricados por IBM en los años 1980. Se diseñan en general para una computación en tiempo real. En un sistema embebido la mayoría de los componentes se encuentran incluidos en la placa base. Existen también plataformas desarrolladas por distintos fabricantes que proporcionan herramientas para el desarrollo y diseño de aplicaciones y prototipos con sistemas embebidos desde ambientes gráficos, algunos ejemplos de estas son: Arduino, RaspberryPi, etc

COMPONENTES:

Tenemos el microprocesador o microcontrolador en la parte central (CPU o capacidad de cómputo) la comunicación puede darse mediante interfaces estándar de cable o inalámbricas, algunos puertos de comunicación pueden ser IP, wi-fi. Tenemos los actuadores que pueden ser motor eléctrico, conmutador tipo relé, salida PWM para control de motor DC, etc.

ARDUINO VS RASPBERRY PI:

Son las placas más populares para hacer trabajos aficionados y profesionales

Ambos se pueden programar usando el lenguaje ensamblador del microcontrolador o microprocesador, C o C++. Arduino y Raspberry Pi están orientados al diseño y desarrollo de aplicaciones y prototipos.

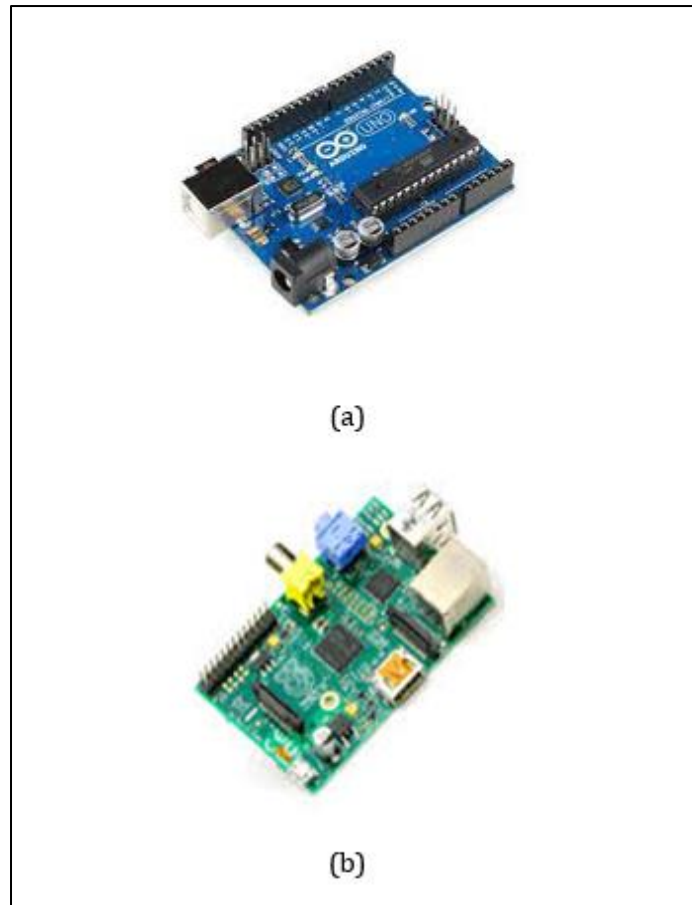


Fig 1. (a) Microcontrolador Arduino UNO (b) Microprocesador RaspberryPi.

ARDUINO:

Arduino es una compañía tecnológica dedicada principalmente al diseño y fabricación de placas de desarrollo hardware con un objetivo principal en mente: el aprendizaje. Se centra en la facilidad y accesibilidad para todos. Contiene un microcontrolador a programar con varios elementos que facilitan la realización de pruebas

Tipos:

Arduino uno

La placa tiene 7 pines de alimentación, 14 pines digitales y 6 pines analógicos programables con el Arduino IDE (Entorno de desarrollo integrado) a través de un cable USB. Puede ser alimentado por el cable USB o por una batería externa de 9 voltios, aunque acepta voltajes entre 7 y 20 voltios. Es el buque insignia de Arduino ya que es la placa más popular, la que

todo el mundo utiliza para iniciarse y la más sencilla de utilizar. Es el punto de partida de muchos entusiastas de la programación electrónica.

Arduino Mega

La placa con el microcontrolador más potente de la familia Arduino. Con 54 pines digitales que funcionan como entrada y salida; 16 entradas analógicas, 7 pines de alimentación, un cristal oscilador de 16 MHz, una conexión USB, un botón de reinicio y una entrada para la alimentación de la placa. Arduino MEGA es la placa que se utiliza cuando Arduino UNO no llega a cubrir las necesidades de un proyecto. Se ha utilizado ampliamente como centro de control y computación en impresoras 3D.

Arduino Esplora

La placa utiliza un microcontrolador Atmega32U4 AVR con oscilador de cristal de 16 MHz y una conexión micro USB capaz de actuar como un dispositivo cliente USB, como un mouse o un teclado.

Arduino Leonardo

La placa tiene 20 pines de entrada/salida digital (de los cuales 7 se pueden usar como salidas PWM y 12 como entradas analógicas), un oscilador de cristal de 16 MHz, una conexión micro USB, un conector de alimentación, un encabezado ICSP y un botón de reinicio.

MICROCONTROLADOR:

Los microcontroladores son circuitos integrados compuestos de entradas salidas, memoria y unidades lógico aritméticas. Son un elemento completo y funcional para realizar operaciones digitales. Realizan menos instrucciones por segundo que un procesador (son mas lentos)

Características de los microcontroladores:

- Son compatibles para reducir el consumo de energía eléctrica
- Estos dispositivos se pueden convertir en una unidad central de procesamiento con pocos circuitos integrados de manera externa
- Poseen una pequeña cantidad de memoria aleatoria
- Cuentan con elementos de entrada y salida como el temporizador y los buses de interfaz
- Disponen de ondas de alta frecuencia, las cuales configuran los impulsos del reloj que se usa para la regularización de todos los métodos donde se aplique

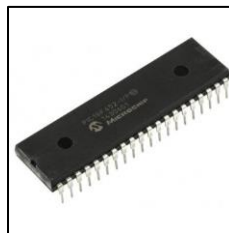


Fig 2. Microcontrolador

Partes de un microcontrolador

- CPU: Unidad central de procesamiento
- Memoria RAM: Memoria de acceso aleatorio, es un espacio de almacenamiento temporal, se usa para hacer cálculos y otras operaciones lógicas
- Memoria ROM: Memoria de solo lectura, se almacenan datos que no cambian, se graban al momento de hacer el chip.
- Líneas de entrada y salida: Las entradas pueden ser analógicas o digitales, usan las salidas para ejecutar acciones programadas

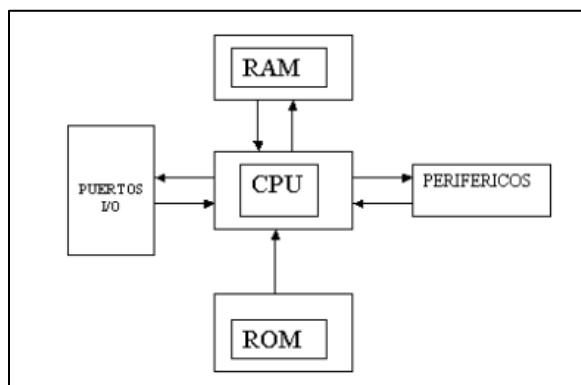


Fig 3. Partes importantes de un microcontrolador.

RASPBERRY PI:

RaspberryPi tiene como objetivo principal la educación y la accesibilidad. Crea placas económicas y bien documentadas. Su producto, las Raspberry Pi, son lo que se conoce como Single-Board Computer. Son ordenadores enteros concentrados en una sola placa. Contienen todos los componentes que requiere un ordenador. Tiene un sistema operativo y herramientas (posee un microprocesador)

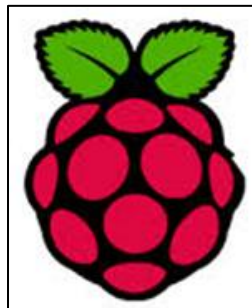


Fig 4. Logo clásico asociado al RaspberryPi.

MICROPROCESADOR:

Un microprocesador es un elemento que realiza operaciones lógico aritméticas. No dispone de entradas y salidas como un microcontrolador. Requiere de más periféricos adicionales para funcionar, como memorias o controladores de bus. Un procesador es el cerebro de un ordenador que consiste básicamente en una unidad aritmética y lógica (ALU), una unidad de control y una matriz de registro. Como su nombre indica, ALU realiza todas las operaciones aritméticas y lógicas sobre los datos recibidos de los dispositivos de entrada o memoria. La matriz de registro consiste en una serie de registros como el acumulador (A), B, C, D, etcétera, que actúa como ubicaciones temporales de memoria de acceso rápido para procesar datos. Por su parte, la unidad de control gestiona el flujo de instrucciones y datos en todo el sistema. Así que básicamente, un microprocesador toma la entrada de los dispositivos conectados, la procesa según las instrucciones dadas en la memoria, y produce la salida.

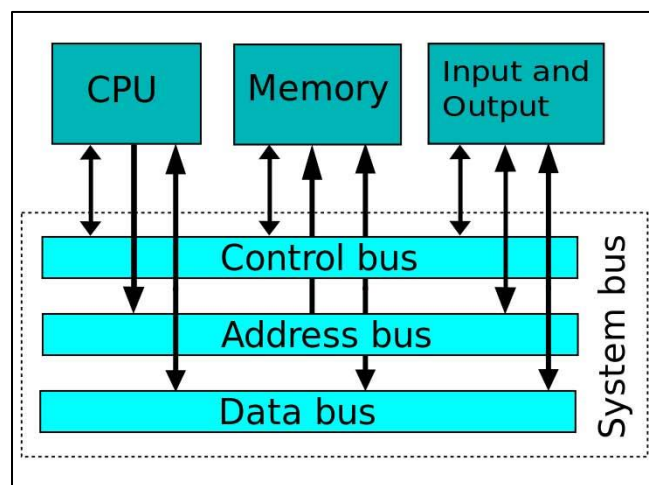


Fig 5. Esquema general de un microprocesador.

VENTAJAS DE ARDUINO:

- Facilidad de manejo
- Muchas guías y documentación en la web
- Necesita poca energía
- Es económico

VENTAJAS DE RASPBERRY PI:

- Potencia
- Rapidez
- Buena capacidad de procesamiento

Aplicativo web

Para el desarrollo de la aplicación web se ha utilizado el editor Sublime Text 3.

Sublime Text 3 es un programa de edición de código que permite ser capaz de escribir código de manera eficiente. Con él es posible hacer todo lo necesario dentro del proceso de codificación, contando con un sólido número de plugins y extras para personalizar la plataforma directamente según los requisitos de cada usuario. Fue creado basado en **Python** y **C++**.

En primer lugar, hay que **descargar Sublime Text** e instalarlo en el ordenador. Una vez hecho esto, al abrir el software se puede ver que es bastante plano y sencillo. El editor depende en gran medida de los paquetes adicionales que se instalen.

Para hacerlo, hay que ir a Tools y hacer clic en Install Package Control. Una vez hecho esto, se puede comenzar a descargar lo que resulte necesario e implementar una nueva interfaz de usuario. Encontrar lo que es mejor para cada usuario acelerará la forma en la que se aprovechará el software para escribir código.

HTML

HTML (Lenguaje de Marcas de Hipertexto, del inglés HyperText Markup Language) es el componente más básico de la Web. Define el significado y la estructura del contenido web. Además de HTML, generalmente se utilizan otras tecnologías para describir la apariencia/presentación de una página web (CSS) o la funcionalidad/comportamiento (JavaScript).

"Hipertexto" hace referencia a los enlaces que conectan páginas web entre sí, ya sea dentro de un único sitio web o entre sitios web. Los enlaces son un aspecto fundamental de la Web. Al subir contenido a Internet y vincularlo a las páginas creadas por otras personas, te conviertes en un participante activo en la «World Wide Web» (Red Informática Mundial).

HTML utiliza "marcas" para etiquetar texto, imágenes y otro contenido para mostrarlo en un navegador Web. Las marcas HTML incluyen "elementos" especiales como <head>, <title>, <body>, <header>, <footer>, <article>, <section>, <p>, <div>, , <video>, , , , y muchos otros.

Un elemento HTML se distingue de otro texto en un documento mediante "etiquetas", que consisten en el nombre del elemento rodeado por "<" y ">". El nombre de un elemento dentro de una etiqueta no distingue entre mayúsculas y minúsculas. Es decir, se puede escribir en mayúsculas, minúsculas o una mezcla.

CSS

Se trata de una tecnología utilizada para dotar de cualidades visuales y estéticas a una página web. Si nunca has tocado esta materia, comprobarás que se trata de una forma analítica, lógica y prácticamente matemática de crear páginas web, pero gracias a ella podemos simplificar la creación de páginas y conseguir exactamente lo que buscamos.

Las siglas CSS (Cascading Style Sheets) significan «Hojas de estilo en cascada» y parten de un concepto simple pero muy potente: aplicar estilos (colores, formas, márgenes, etc.) a uno o varios documentos (generalmente documentos HTML, páginas webs) de forma masiva.

Se le denomina estilos en cascada porque se aplican de arriba a abajo (siguiendo un patrón denominado herencia que trataremos más adelante) y en el caso de existir ambigüedad, se siguen una serie de normas para resolverla.

La idea de CSS es la de utilizar el concepto de separación de presentación y contenido, intentando que los documentos HTML incluyan sólo información y datos, relativos al significado de la información a transmitir (el contenido), y todos los aspectos relacionados con el estilo (diseño, colores, formas, etc.) se encuentren en un documento CSS independiente.

JAVASCRIPT

Javascript es un lenguaje poderoso, capaz de aportar soluciones eficaces en la mayoría de los ámbitos de la tecnología.

Es especialmente importante porque es el único lenguaje de programación que entienden los navegadores, con el que se desarrolla la parte de la funcionalidad frontend en sitios web y aplicaciones web modernas. Pero también es fundamental en muchos otros tipos de desarrollos. Sus usos más importantes son los siguientes:

- Desarrollo de sitios web del lado del cliente (frontend, en el navegador).
- Desarrollo de todo tipo de aplicaciones gracias a la plataforma NodeJS.
- Desarrollo de aplicaciones para dispositivos móviles, híbridas o que compilan a nativo.
- Desarrollo de aplicaciones de escritorio para sistemas Windows, Linux y Mac, pudiendo escribir un código compatible con todas las plataformas.

Por tanto, podemos considerar a Javascript el lenguaje universal, pues es el que más tipos de aplicaciones y usos que puede abarcar en la actualidad. Es por ello que resulta un lenguaje muy recomendable para aprender, ya que nos ofrece capacidades para usarlo en todo tipo de proyectos, siendo que algunos de ellos son parcela exclusiva de Javascript.

La aplicación web ha sido desarrollada de manera en que tenga un espacio en la nube utilizando Firebase de Google.

4. Desarrollo del proyecto

Para el proyecto construiremos un sistema mecánico para la calibración del equipo y movimiento de la jeringa aplicando IoT.

4.1 Componentes y presupuesto estimado

Componentes	Precio estimado (soles)
Motor CD-Rom	90
Driver Puente H L298N	15
Raspberry Pi	200

Jeringas	5
Sistema mecánico	10
Protoboard + cables	5
Total (soles): 325	

3.1 Sistema mecánico

3.1.1 Jeringa

Nos permite ingresar el fluido deseado al paciente con la velocidad de flujo y cantidad requerida previamente. En este proyecto utilizamos una jeringa comercial estándar de 5ml

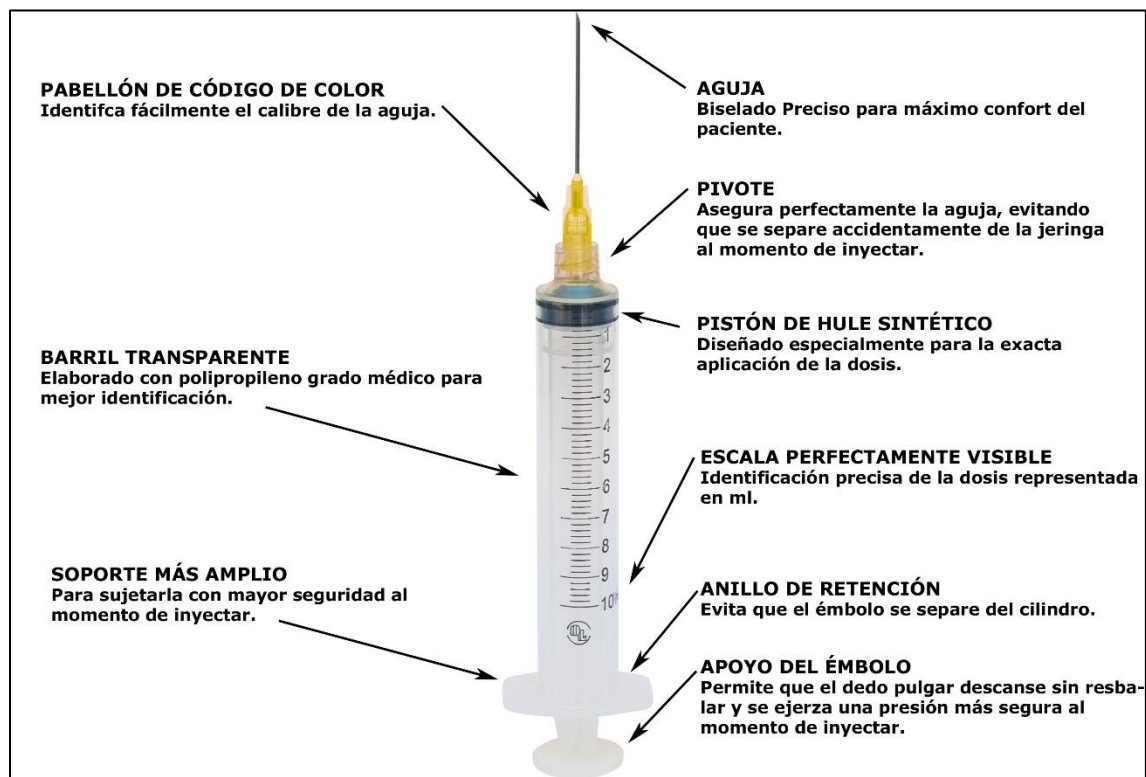


Fig 6. Partes de una jeringa comercial estándar.

Es importante calibrar la jeringa antes de usarse ya que suelen presentarse ligeras imperfecciones o irregularidades de fábrica.

Para realizar la calibración de la jeringa se realizará lo siguiente:

- *Carga de la jeringa*: en este proceso vamos cargando la jeringa de medio mililitro (0.5 ml) en medio mililitro.
- *Pesado del líquido*: luego de la carga, se hace el pesado del líquido extraído con la balanza analítica. El líquido extraído en esta calibración se hizo con agua destilada cuya densidad a 20 °C (que fue la temperatura del ambiente en el momento de la calibración)

es de 0.99820 kg/l. En cada una de las pesadas se hace el tarado del recipiente donde pusimos el agua que en esta ocasión fue un vaso de plástico.

- *Corrección del volumen:* aquí utilizamos la densidad para convertir lo medido en la balanza analítica (masa en gramos) en unidades de volumen.
- *Cálculo de error:* comparamos el volumen extraído en la jeringa con el volumen hallado con la corrección anterior.

Posterior a este proceso de obtención de datos, se hace un análisis de ellos para poder hallar el error en el volumen de líquido expulsado por la jeringa.

VOLUMEN (ml)	MASA PESADA (g)	VOLUMEN CORREGIDO (ml)	ERROR
0.2	0.2040	0.2044	-0.0044
0.4	0.3968	0.3975	0.0025
0.6	0.5905	0.5916	0.0084
0.8	0.8191	0.8206	-0.0206
1.0	1.0297	1.0316	-0.0316
1.2	1.2210	1.2232	-0.0232
1.4	1.4947	1.4974	-0.0974
1.6	1.6534	1.6564	-0.0564
1.8	1.8313	1.8346	-0.0346
2.0	2.0124	2.0160	-0.0160
2.2	2.2145	2.2185	-0.0185
2.4	2.4252	2.4296	-0.0296
2.6	2.6282	2.6329	-0.0329
2.8	2.8357	2.8408	-0.0408
3.0	3.0512	3.0567	-0.0567
3.2	3.2116	3.2174	-0.0174
3.4	3.4137	3.4199	-0.0199
3.6	3.6321	3.6386	-0.0386
3.8	3.8356	3.8425	-0.0425
4.0	4.0191	4.0263	-0.0263
4.2	4.2243	4.2319	-0.0319
4.4	4.4010	4.4089	-0.0089
4.6	4.6173	4.6256	-0.0256
4.8	4.8443	4.8530	-0.0530
5.0	5.0845	5.0937	-0.0937
		PROMEDIO	-0.0324
		DESVIACION ESTANDAR	0.0251

Tabla 1. Calibración de la jeringa comercial estándar de 5ml antes de utilizarlo.

Finalmente, con todos los datos obtenidos y analizados, tenemos el volumen total de la jeringa con su margen de error asociado:

$$V = 5.000 \pm 0.025 \text{ ml}$$

DISTANCIA AVANZADA POR ROTACIÓN

	AVANCE DEL TORNILLO PARA UNA ROTACIÓN (mm)
	2.8
	2.7
	2.7
	2.7
	2.65
	2.8
	2.85
	3.45
	3.1
	3.6
	2.8
	3.2
PROMEDIO	2.95
DESVIACION ESTANDAR	0.32

Tabla 2. Distancia lineal asociada a una rotación del tornillo de avance.

En consecuencia, una rotación del tornillo de avance es equivalente a un avance de la jeringa:

$$D = 2.95 \pm 0.32 \text{ mm}$$

3.1.2 Carcasa para la bomba de jeringa

Esta carcasa nos servirá de soporte para fijar nuestro sistema de control (motor cdrom) y jeringa comercial. Las piezas fueron hechas en una impresora 3D, en la cual las partes de esta fueron convertidas de extensión. STL a extensión. gcode. Asimismo, se utilizó el material PLA

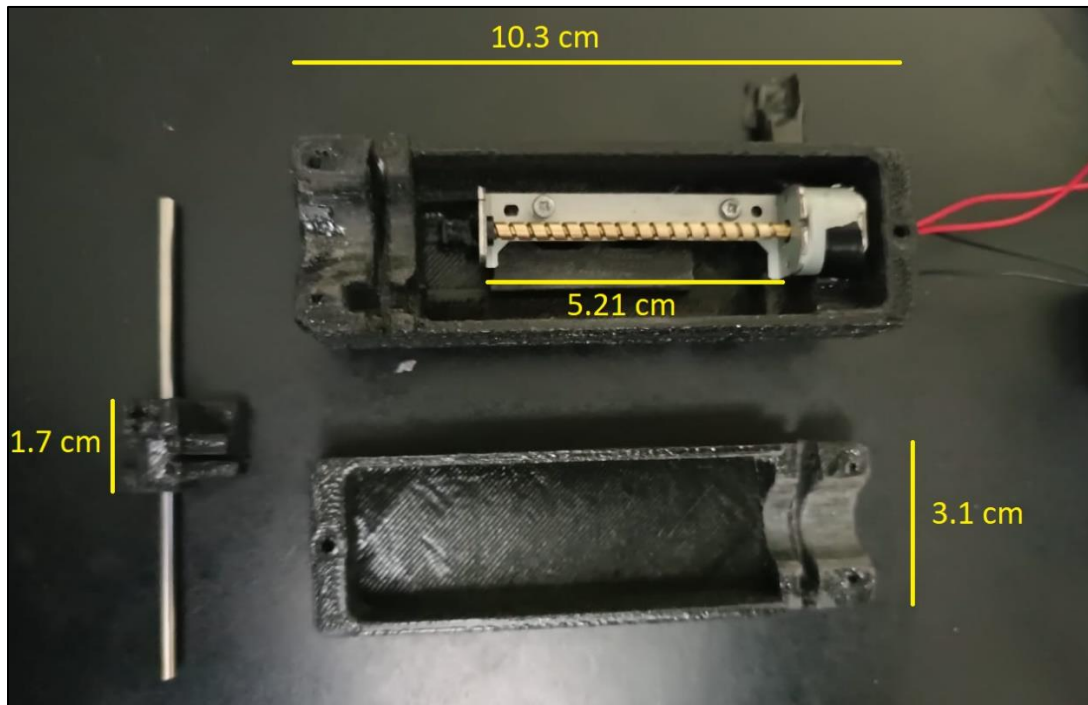


Fig 7. Dimensiones importantes de la estructura mecánica de soporte impresa en 3D con material PLA

3.2 Motor DC, CD-ROM

El motor de corriente continua a paso nos permite darle movimiento a todo el sistema mecánico, en consecuencia, brindarle movimiento a la jeringa. El motor utilizado es un motor cd-rom o “stepper motor cd-rom” el cual trabaja con 5V, un ángulo de paso de 18° y una resistencia interna de 10 ohm con 7% de incertidumbre.



Fig 8. Motor DC Bipolar sacada de un drive CD-ROM

3.2.1 Tornillo de avance

Con esta varilla roscada convertimos el movimiento rotacional de nuestro motor en movimiento lineal o en línea recta para poder accionar la bomba de jeringa. Este tornillo de avance posee 5.21 cm de largo.

3.3 Esquema del proyecto

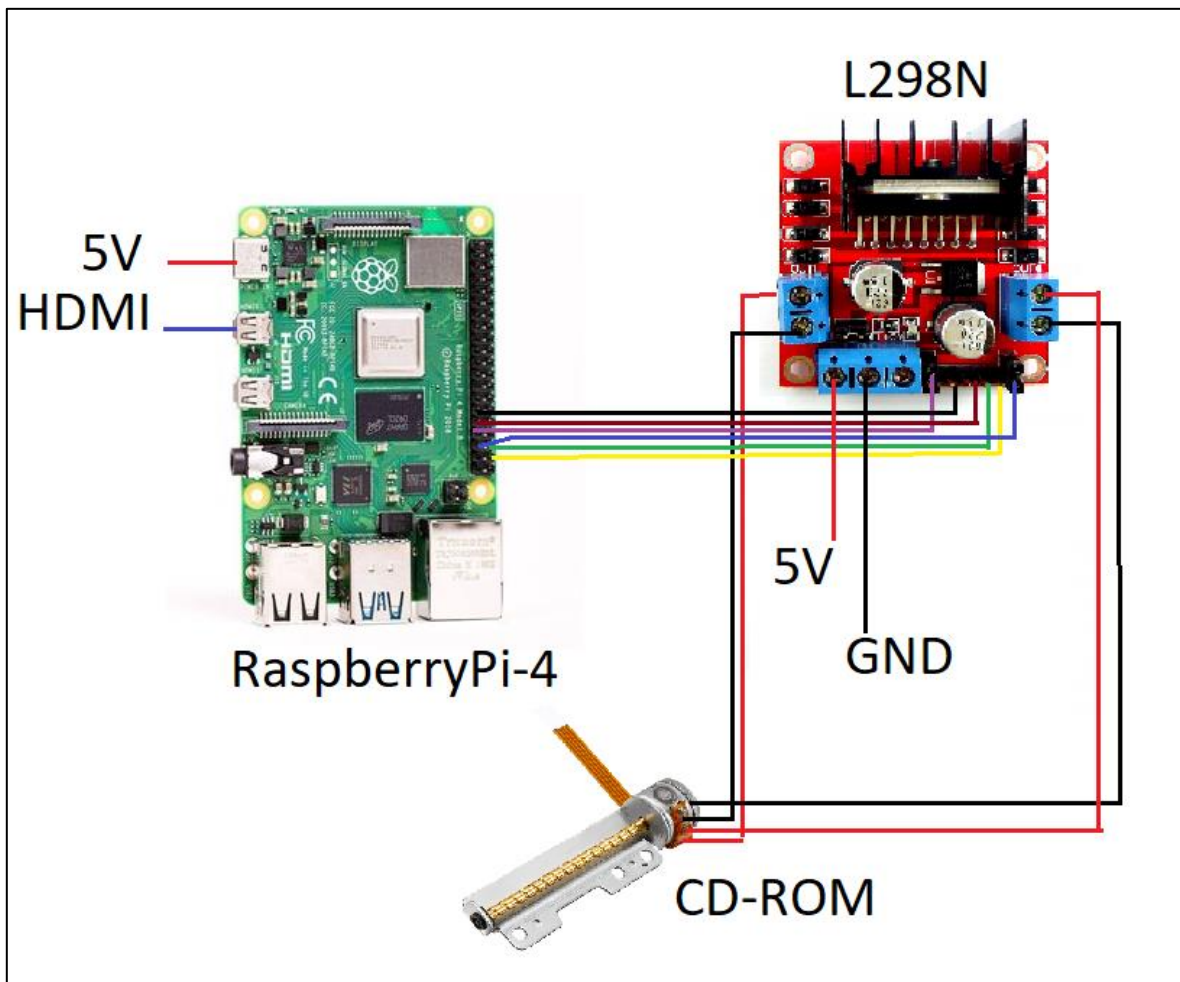


Fig 9. Esquema de conexiones del Hardware utilizado.

3.4 Procedimiento y conexiones.

En nuestro sistema necesitamos controlar y calibrar con buena precisión la cantidad y velocidad del fluido inoculado desde la jeringa hacia un agente externo. Por tanto, se realizó el estudio del funcionamiento interno del motor a paso CD-ROM, el cual es un motor a paso rotativo BIPOLAR. Dicho motor tiene 2 bobinas, que está ON y cambia de dirección sincronizada para hacer que el rotor corra hacia adelante / hacia atrás por pulso emitido desde el RaspberryPi4. Cada pulso girará el rotor cierto movimiento, (llamado paso) de esta manera el rotor puede ser controlado su posición por pulso.

Previamente se hace una prueba de continuidad del CD-ROM para identificar los pines asociados a cada una de las 2 bobinas que presenta el motor Bipolar.

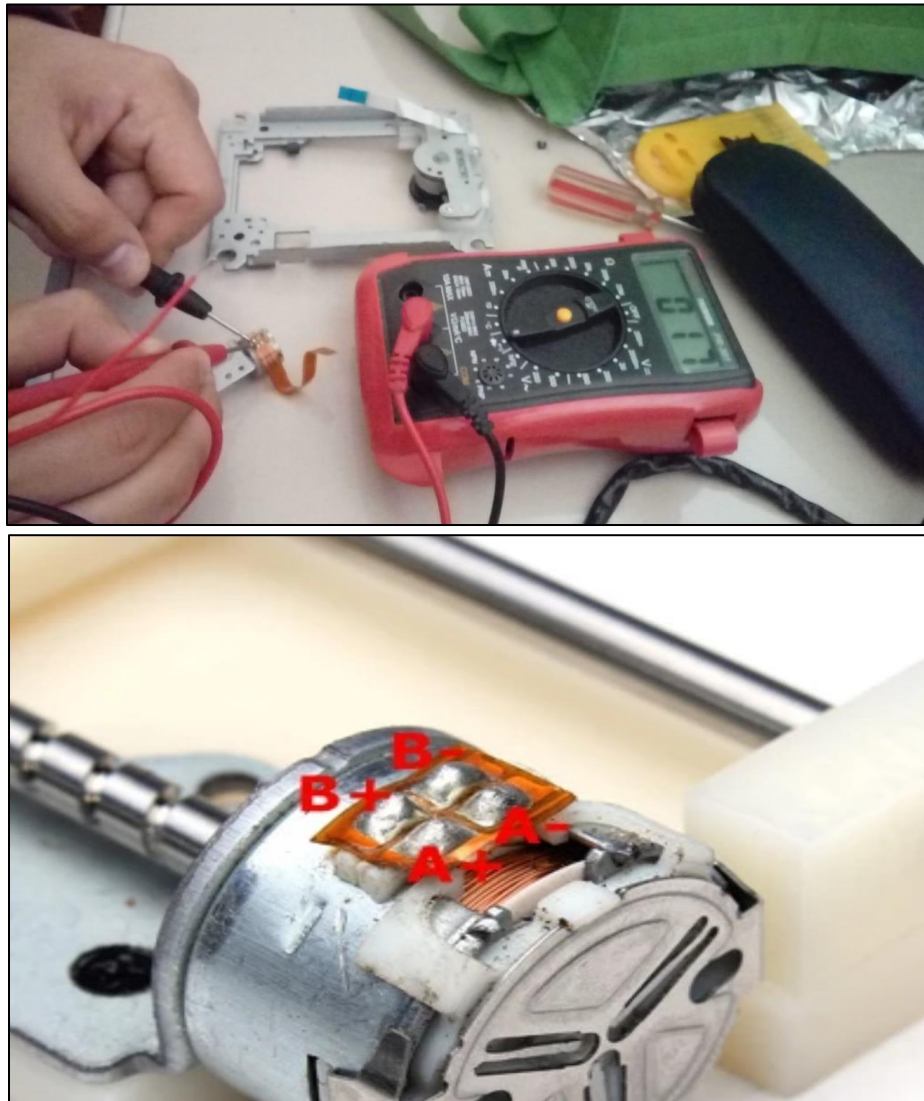


Fig 10. Identificación de la polaridad del motor bipolar mediante una prueba de continuidad utilizando el multímetro.

Con el puente H L298N se controla el pulso asociado a cada bobina del CD-ROM, siendo las conexiones de la bobina 1 a las entradas al OUT1 y OUT2, así como para la bobina 2, las entradas OUT3 y OUT4 del puente H.

Básicamente, hay muchos patrones de pulso para el motor paso a paso, como el impulso de una fase, de dos fases (paso completo), medio paso, entre otros. En este proyecto hemos desarrollado el control bajo impulso de paso completo, el cual queda descrito por la siguiente tabla:

	1	2	3	4
A	HIGH	LOW	LOW	HIGH
~A	LOW	HIGH	HIGH	LOW
B	HIGH	HIGH	LOW	LOW
~B	LOW	LOW	HIGH	HIGH

Tabla 3. Lógica de control digital para cada bobina A y B asociado a la rotación del motor.

El cual describe el siguiente movimiento:

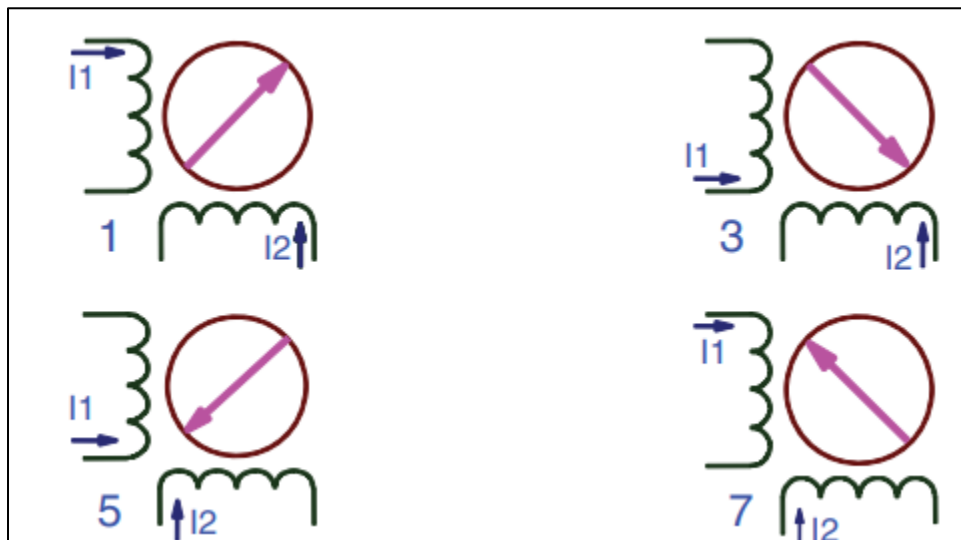


Fig 11. Secuencia de paso completo para un motor bipolar.[4]

3.5 Drive controlador Puente H.

Este es un driver controlador de motores, específicamente se planea utilizar el L298N el cual nos permite controlar un motor DC debido a la cantidad de corriente que demanda el uso de un motor y a su vez proteger al microprocesador RaspberryPi. Este módulo trabaja con 5V y 1A. Se utiliza los pines digitales del RaspberryPi4:

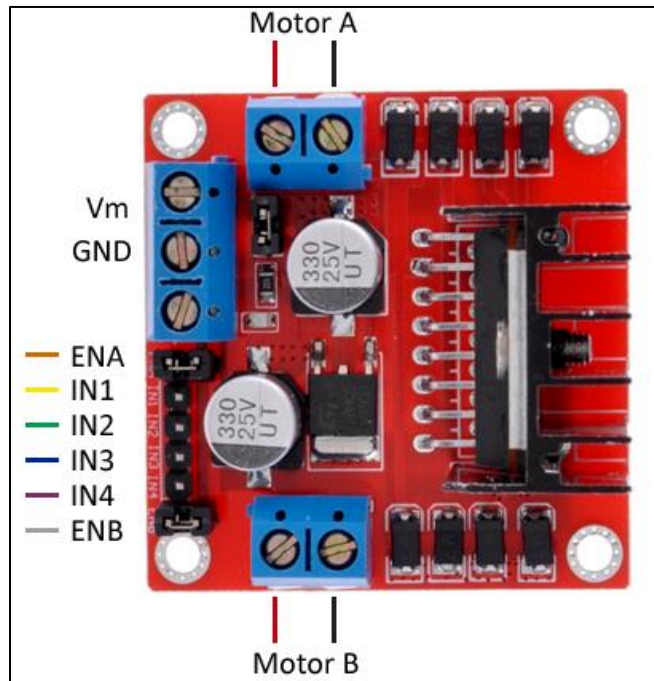


Fig 12. Drive L298N Puente H.

Pines digitales utilizados del RaspberryPi4:

$pinA1 = 29$
 $pinA2 = 31$
 $pinENA = 33$ (PWM)
 $pinB1 = 36$
 $pinB2 = 38$
 $pinENB = 35$ (PWM)

4 Aplicativo Web y Base de datos.

Diseñamos un aplicativo web interactiva utilizando programación en JavaScript y lenguaje HTML desde el cual podemos controlar el accionamiento de la bomba de jeringa, bajo 3 operaciones secuenciales: Calibración, Velocidad, Volumen a inocular.

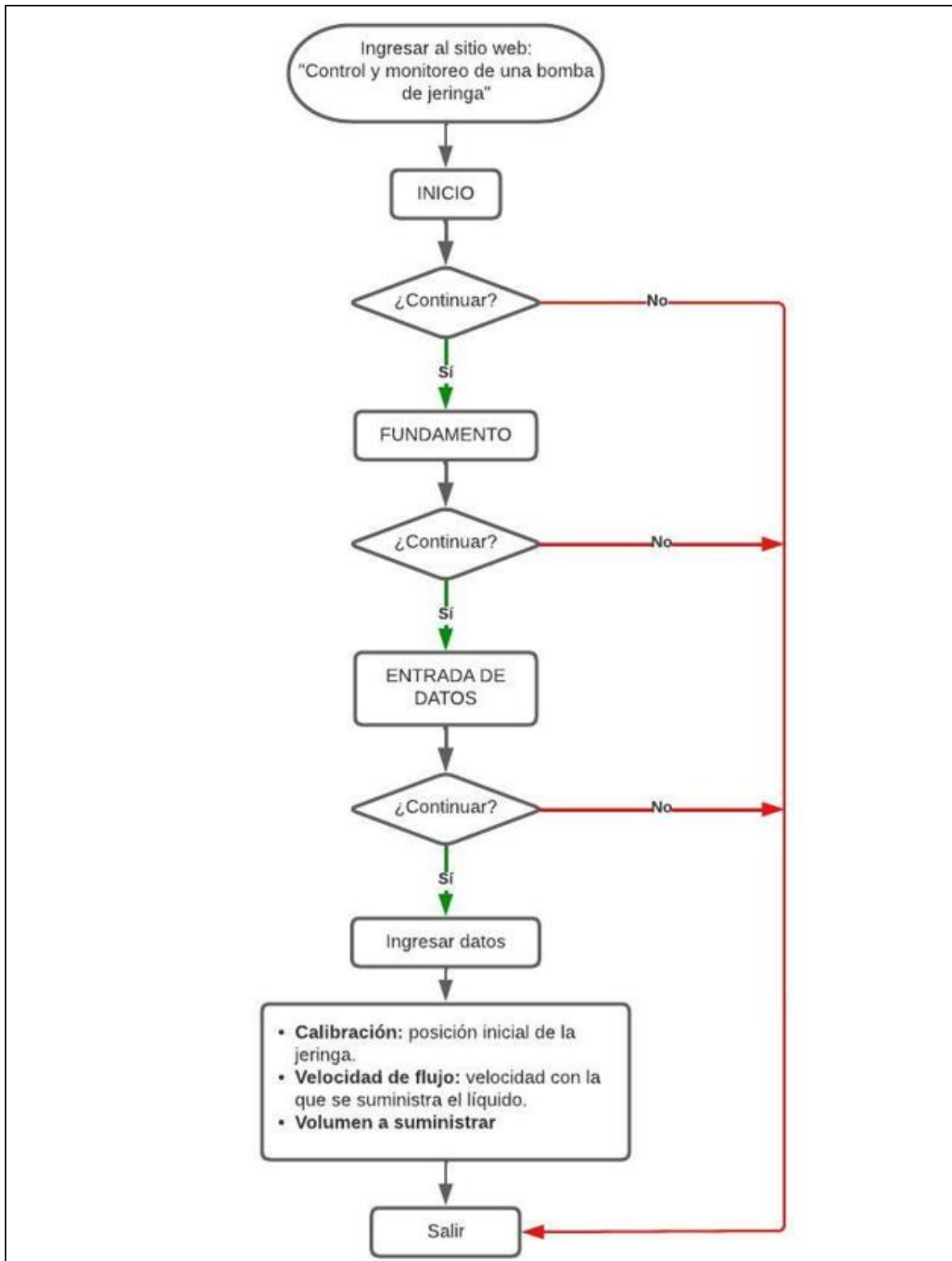


Fig 13. Diagrama de flujo general del aplicativo web.

El enlace para la página es:
<https://iot-bombadejeringa.web.app>

CONTROL Y MONITOREO DE UNA BOMBA DE JERINGA

Inicio
Fundamento
Entrada de datos

Proyecto de Instrumentación Electrónica

¿Qué es una bomba de jeringa?

Una bomba de jeringa es un dispositivo de manipulación de líquidos que se utiliza para suministrar una cantidad precisa de líquido en entornos industriales y de investigación muy sensibles. Las aplicaciones de una bomba de jeringa pueden variar desde la química y la biología en el campo de la investigación hasta el procesamiento de alimentos y los requisitos de microdispensación para aplicaciones industriales.

Objetivo del proyecto

Debido al contexto en que nos encontramos, como es el de una pandemia, así como la poca accesibilidad a algunas zonas rurales, hacen que la aplicación de la telemedicina se haga una necesidad más frecuente. Es por ello, que este proyecto pretende ayudar a entregar fluidos como nutrientes y medicamentos al cuerpo de un paciente, agente externo y/o equipo que lo requiera en un ambiente controlado.



CONTROL Y MONITOREO DE UNA BOMBA DE JERINGA

Inicio
Fundamento
Entrada de datos

Proyecto de Instrumentación Electrónica

Tutorial sobre el manejo adecuado de la bomba de infusión y calibración:



Para el volumen experimental tendremos en consideración que 3.82 cm es la distancia desde la posición inicial (POSICIÓN X=0) hasta el punto donde se ha expulsado todo el líquido (POSICIÓN X=5). También se a considerado que 0.82 cm es la distancia que hay entre línea y línea que determinan 1 ml en la jeringa comercial estándar de 5ml de volumen.

VELOCIDAD DE FLUJO = Valores experimentales/segundos

0.82 cm = 1 ml
3.82 cm = Volumen exp.
= Volumen experimental = 4.85 ml

	t1	t2	t3
t1	2.75	14.55	26.85
t2	2.75	14.55	26.85
t3	2.81	15.05	26.35
Temps tot	5.56	29.6	53.2
VELOCIDAD DE FLUJO (ml/s)	0.09	0.02	0.09

Donde t1, t2 y t3 son los tiempos medidos en los que el carrete se mueve desde la POSICIÓN X=0 hasta la POSICIÓN X=5, asociados a un volumen inicial de 0ml y un volumen teórico final de 5ml respectivamente.

CONTROL Y MONITOREO DE UNA BOMBA DE JERINGA

Inicio
Fundamento
Entrada de datos

Proyecto de Instrumentación Electrónica

Posicionamiento/Calibración ON

Velocidad de flujo (ml/s) v1 = 167 ml/s

Volumen a suministrar (ml): 1 ml

Fig 14. Se muestra las 3 caras desarrolladas en el aplicativo web para el control de la bomba de infusión.

Proceso de control en el App Web

$$VELOCIDAD DE FLUJO = Volumen\ experimental / t_{prom}$$

Para el volumen experimental tendremos en consideración que 3.82 cm es la distancia desde la posición inicial (POSICIÓN X=0) hasta el punto donde se ha expulsado todo el líquido (POSICIÓN X=5). También se ha considerado que 0.82 cm es la distancia que hay entre línea y línea que determinan 1 ml en la jeringa comercial estándar de 5ml de volumen:

$$\begin{array}{lcl} 0.82\text{ cm} & \text{————} & 1\text{ ml} \\ 3.82\text{ cm} & \text{————} & Volumen\ exp. \\ \Rightarrow Volumen\ experimental & = & 4.85\text{ ml} \end{array}$$

Con estos datos previos tendremos la obtención del parámetro Velocidad de flujo (ml/s) para diferentes velocidades (v1, v2 y v3) en las que se mueve el tornillo de avance.

	v1	v2	v3
t1	2.79	14.55	26.01
t2	2.75	14.53	26.34
t3	2.81	15.03	26.15
Tprom (s)	2.78	14.7	26.16
VELOCIDAD DE FLUJO (ml/s)	1.67	0.32	0.18

Tabla 4. Cálculo de velocidades en ml/s.

Donde t1, t2 y t3 son los tiempos medidos en los que el carrito se mueve desde la POSICIÓN X=0 hasta la POSICIÓN X=5, asociados a un volumen inicial de 0ml y un volumen teórico final de 5ml respectivamente.

Este aplicativo web nos permite en primera instancia calibrar la posición inicial de la jeringa ya cargada, posicionándola en un punto definido como X=0 en la cual la jeringa se encuentra completamente llena, es decir, conteniendo 5ml de sustancia. Para este proceso se debe activar en “ON” el Posicionamiento/Calibración y a su vez indicar en el bloque de Volumen a suministrar(ml) que se ejecutará una calibración seleccionando “OFF(CALIBRANDO)”

Posicionamiento/Calibracion

ON ▼

Ejecutar

ON

OFF

This screenshot shows a software interface with a light blue background and a dashed border. It contains a label 'Posicionamiento/Calibracion', a dropdown menu currently showing 'ON', an 'Ejecutar' button, and a secondary menu with 'ON' and 'OFF' options.

Luego de ello, se procede a definir el flujo (velocidad del fluido) con el cual se desea inocular la sustancia, para esto se ha definido y calculado 3 tipos de velocidades:

Velocidad de flujo (ml/s)

v1 = 1.67 ml/s ▼

Ejecutar

v1 = 1.67 ml/s

v2 = 0.32 ml/s

v3 = 0.18 ml/s

This screenshot shows a software interface with a light blue background and a dashed border. It contains a label 'Velocidad de flujo (ml/s)', a dropdown menu showing 'v1 = 1.67 ml/s', an 'Ejecutar' button, and a list of three options: 'v1 = 1.67 ml/s', 'v2 = 0.32 ml/s', and 'v3 = 0.18 ml/s'.

Finalmente se selecciona la cantidad de volumen a suministrar, para ello se a definido previamente 5 puntos de volumen, los cuales van asociados al tipo de jeringa utilizada, la cual es una jeringa comercial estándar de 5ml:

Volumen a suministrar (ml):

- 1 ml
- 2 ml
- 3 ml
- 4 ml
- 5 ml
- OFF(CALIBRANDO)

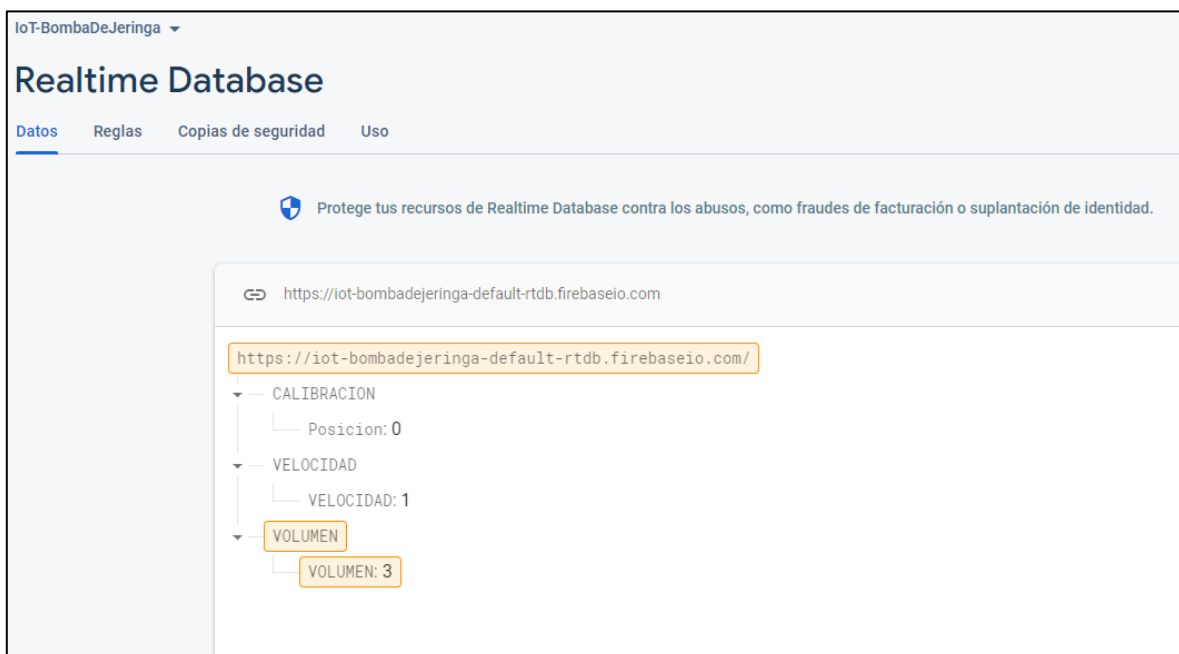


Fig 15. En este cuadro se muestra una fotografía instantánea de la base de datos Firebase en la cual se ejecuta una inoculación de 3ml de sustancia a una velocidad $V1=1.67\text{ml/s}$.

5 Diagrama de Gantt

N°	Actividades por componentes	Semanas															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	Análisis de la viabilidad del proyecto																
1.1	Recopilación de información bibliográfica	x															
1.2	Estudio de los componentes electrónicos y mecánicos a utilizar		x														
1.3	Elaboración de presupuesto		x														
1.4	Esquematación del circuito de control de la bomba de jeringa			x													
2	Transmisión de datos a través de la nube																
2.1	Estudio de la plataforma de Firebase para el envío y recepción de datos				x												
2.2	Programación del aplicativo web para el control de la bomba					x	x										
2.3	Ensayo del funcionamiento del motor DC remotamente							x									
3	Manejo y estudio de los componentes electrónicos																
3.1	Programación y uso del RaspberryPi								x								
3.2	Programación, control y conexiones del CD-ROM									x							
3.3	Uso del puente H L298N para el control del motor									x							
4	Diseño mecánico																
4.1	Impresión 3D de las piezas mecánicas										x						
4.2	Esquematación y armado del sistema mecánico											x	x				
5	Construcción del prototipo de bomba de jeringa																
5.1	Establecer parámetros de acuerdo a lo requerido													x	x	x	x
5.2	Ejecutar pruebas															x	x

Fig. 16. Diagrama de Gantt candelarizado que detalla el desarrollo del proyecto a lo largo de 16 semanas.

6 Resultados

Se muestra los valores de volúmenes obtenidos de acuerdo a ordenes solicitadas y ejecutadas desde el aplicativo web para 5 valores de volumen, que van desde 1ml hasta 5ml (valor máximo) utilizando una velocidad de inoculación de 1.67 ml/s.

VOLUMEN (ml)	VOLUMEN EXPERIMENTAL (ml)	ERROR
1	0.88	0.12
2	1.89	0.11
3	2.84	0.26
4	3.88	0.12
5	4.86	0.14

Tabla 5. Volúmenes inoculados obtenidos a una velocidad de flujo de 1.67 ml/s

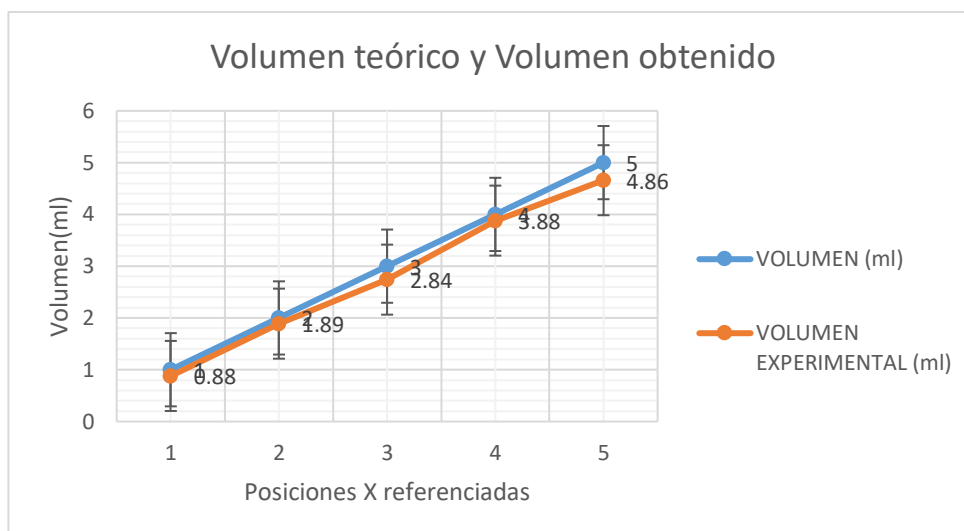


Fig. 17. Gráfico que compara el volumen teórico vs el volumen obtenido por el sistema desarrollado.

Determinamos las longitudes recorridas por el soporte 3D que sostiene la base de la jeringa comercial de 5ml asociados a los pulsos e iteraciones ejecutados desde el aplicativo web previamente programado en el microprocesador Raspberry en lenguaje Python. Obteniendo así unas posiciones promedias que nos ayudaron a determinar las velocidades, recorrido completo asociado al volumen total de infusión y finalmente la posición inicial de calibrado.

	POSICIÓN 1	POSICIÓN 2	POSICIÓN 3	POSICIÓN 4	POSICIÓN 5
	X=1	X=2	X=3	X=4	X=5
	0.74	1.55	2.24	3.18	3.8
	0.70	1.58	2.25	3.21	3.82
	0.73	1.53	2.23	3.15	3.84
	0.71	1.57	2.26	3.14	3.83
	0.72	1.50	2.27	3.23	3.84
PROMEDIO	0.72	1.55	2.25	3.18	3.82

Tabla 6. Longitudes recorridas por el soporte 3D que sostiene la base de la jeringa ejecutados desde el aplicativo web.

7 Discusión de resultados.

- La determinación del punto X=0 (punto de calibración) se halló a partir de las medidas de las dimensiones del tornillo de avance 5.21 cm, así como también de la medida de la longitud extendida de la jeringa, el cual daba cuenta del volumen total de 5ml.

- Para el cálculo del flujo o velocidad de infusión de la jeringa, se realizó muchas mediciones de tiempo asociado al recorrido del soporte de la base de la jeringa, para el cual se contaba con una distancia fija de 3.82cm asociado a un volumen total experimental de 4.85 ml.
- Respecto al código y control del motor bipolar CD-ROM asociado a la velocidad o flujo de infusión. Se desarrolló un algoritmo que controla eficientemente el tiempo de delay que está presente en el movimiento del rotor conectado al tornillo de avance, el cual es descrito por la tabla lógica (Tabla 3.) y se calculó el número de iteraciones asociado al pulso que nos permitió definir y controlar el avance del tornillo de acuerdo a la cantidad de volumen que se desee suministrar.
- Se presentó un problema mecánico al momento de ejecutar los movimientos de infusión con la jeringa comercial estándar de 5ml. En la cual el motor CD-ROM demandaba un esfuerzo adicional, ocasionando un sobrecalentamiento del motor bipolar, así como el tornillo de avance, en consecuencia, la pieza 3D de material PLA que hace contacto directo con el tornillo se avance sufría los efectos de la temperatura, deformándose e impidiendo un movimiento adecuado con la jeringa colocada. Por tanto, se ha pensado que esto podría solucionarse, utilizando otro material para la fabricación 3D de dicha pieza mecánica, la cual debe soportar el calentamiento del motor.
- Desarrollamos una manera alternativa de medir la eficiencia de la bomba de jeringa, para la cual se tomó medidas con un vernier de la longitud de la jeringa asociada al desplazamiento de 1ml de volumen. Y a su vez, se tomó medición del avance en longitud de la pieza mecánica de soporte al recibir la orden desde el aplicativo web. Obteniendo así una equivalencia asociada al volumen que desplazaría la jeringa comercial según el código y sistema desarrollado.
- El tiempo de retardo entre la ejecución de la orden desde el aplicativo web hasta obtener respuesta a esta orden en el hardware es de aproximadamente 3.45 segundos. Esto puede deberse a la velocidad de internet, así como a la respuesta de ejecución del código Python asociado al microprocesador RaspberryPi.

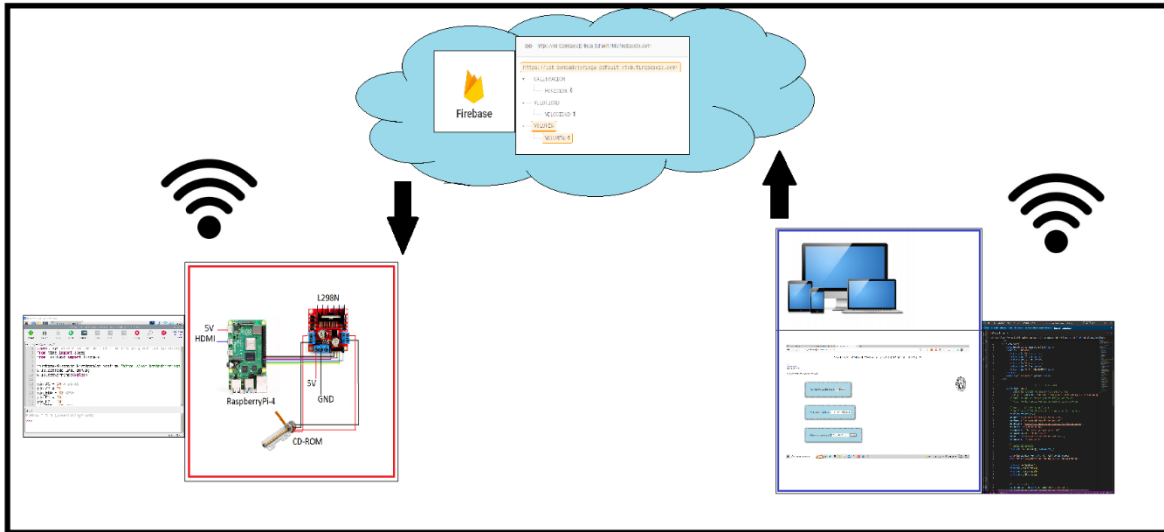


Fig 17. Esquema general del funcionamiento del proyecto: Hardware + Software + Base Datos

8 Conclusiones.

- Obtuvimos una velocidad de respuesta de aproximadamente **3.45 ± 0.04 segundos** para la ejecución de cada orden relacionada a la infusión de determinado volumen de líquido o sustancia.
- Ejecutamos la orden desde el aplicativo web de inocular 1ml de líquido, obteniendo como resultado **0.88 ± 0.07 ml** de líquido, con un error de 0.12. Asimismo, para volúmenes de 2, 3, 4 y 5ml. con un error promedio inferior a 0.15.
- Controlamos eficientemente la velocidad de inoculación de fluido, determinando así 3 tipos de velocidades arbitrarias: **$v_1 = 1.67$ ml/s, $v_2 = 0.32$ ml/s, $v_3 = 0.18$ ml/s.**
- Calibramos previamente la jeringa antes de utilizarla, obteniendo un volumen de **$V = 5.000 \pm 0.025$ ml**. Así como también se desarrolló código para encontrar su punto inicial de calibrado después de infundir el máximo de su capacidad 5ml.
- Desarrollamos un aplicativo web de dominio público con los comandos necesarios de control como: calibración, velocidad de flujo y volumen a inocular. A su vez se estableció correctamente la comunicación de la base de datos con el hardware y con el aplicativo web.

Referencias

- [1] M. R. Islam, R. Zahid Rusho, and S. M. Rabiul Islam, "Design and implementation of low cost smart syringe pump for telemedicine and healthcare," *1st Int. Conf. Robot. Electr.*

Signal Process. Tech. ICREST 2019, pp. 440–444, 2019, doi: 10.1109/ICREST.2019.8644373.

- [2] K. Akash, M. P. Kumar, N. Venkatesan, and M. Venkatesan, “A single acting syringe pump based on Raspberry Pi - SOC,” *2015 IEEE Int. Conf. Comput. Intell. Comput. Res. ICCIC 2015*, pp. 1–3, 2016, doi: 10.1109/ICCIC.2015.7435694.
- [3] S. Dubey, A. Gambhir, S. K. Jain, A. V. Jha, A. Jain, and S. Sharma, “IoT application for the design of digital drug administration interface,” *IEEE Int. Conf. Information, Commun. Instrum. Control. ICICIC 2017*, vol. 2018-January, pp. 1–5, 2018, doi: 10.1109/ICOMICON.2017.8279157.
- [4] D. Maillefer, S. Gamper, B. Frehner, P. Balmer, H. Van Lintel, and P. Renaud, “A high-performance silicon micropump for disposable drug delivery systems,” *Proc. IEEE Micro Electro Mech. Syst.*, pp. 413–417, 2001, doi: 10.1109/MEMSYS.2001.906566.
- [5] M. S. V. Appaji, G. S. Reddy, S. Arunkumar, and M. Venkatesan, “An 8051 Microcontroller based Syringe Pump Control System for Surface Micromachining,” *Procedia Mater. Sci.*, vol. 5, pp. 1791–1800, 2014, doi: 10.1016/j.mspro.2014.07.391.
- [6] G. Fuchs *et al.*, “INTERFACING THE DIGITAL DOMAIN WITH AN ANALOG WORLD Course Manual,” *Sigcse’02*, vol. 3, no. 3, pp. 257–261, 2015, [Online]. Available: <http://www.econ.yale.edu/~dirkb/teach/pdf/akerlof/themarketforlemons.pdf%5Cnhttp://www.atmel.com/images/doc8017.pdf%5Cnhttp://sine.ni.com/nips/cds/view/p/lang/en/nid/14132%5Cnhttp://www.nature.com/doifinder/10.1038/nmeth929%5Cnhttp://classroompresenter.cs.washing>.
- [7] Extraído de: [Syringe pumps and microfluidics - Elveflow](#).