

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA NO INVASIVO DE MONITOREO DE LA PRESIÓN ARTERIAL Y LA FRECUENCIA CARDÍACA.

SISTEMA ELECTRÓNICO DE CONTROL PARA EL TENSÍOMETRO

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN**

VERONICA DALILA GUERRERO GUASHPA

veronica.guerrero@epn.edu.ec

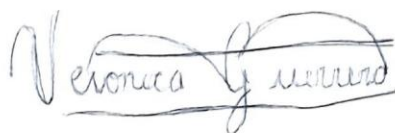
DIRECTOR: EDUARDO ÁVALOS, PH.D.

eduardo.avalos@epn.edu.ec

Distrito Metropolitano de Quito, Abril 2023

CERTIFICACIONES

Yo, VERONICA DALILA GUERRERO GUASHPA declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



Srta. Verónica Dalila Guerrero Guashpa

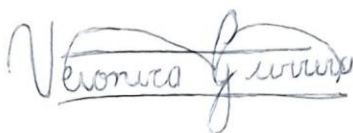
Certifico que el presente trabajo de integración curricular fue desarrollado por VERONICA DALILA GUERRERO GUASHPA, bajo mi supervisión.



Dr. Eduardo Fausto Ávalos Cascante
DIRECTOR

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.



Srta. Verónica Dalila Guerrero Guashpa.



Dr. Eduardo Fausto Ávalos Cascante

INDICE DEL CONTENIDO

CERTIFICACIONES	I
DECLARACIÓN DE AUTORÍA	II
RESUMEN	V
ABSTRACT	VI
1. INTRODUCCION	1
1.1 Objetivo General	1
1.2 Objetivos específicos.	1
1.3 Alcances	2
1.4 Marco teórico	3
1.4.1 Presión Arterial	3
1.4.2 Medición automatizada de la presión	5
1.4.3 Equipo para monitorización ambulatoria de presión.	5
1.4.3.1 Diseño de hardware general de un tensiómetro digital de muñeca...6	
1.4.3.2 Automatización del sistema de control para medición de presión.....8	
1.4.3.3 Métodos de calibración para sensores de presión con salida electrica.....8	
1.4.3.4 Método de estimación de presión arterial en dispositivos digitales. 10	
1.5 Elementos de Hardware utilizados para el sistema de control e Interfaz .12	
1.6 Elementos de hardware utilizados para el sistema neumático	14
1.7 Elementos de Software	15
1.7.1 Elementos de Software para programar ESP32.....	15
1.7.1.1 Plataforma de desarrollo Arduino	15
1.7.2 Elementos de Software para el envío y recepción de datos	16
1.7.2.1 Formato tipo JSON	16
1.7.2.2 Protocolo de transferencia HTTP	16
1.7.2.3 API REST.....	16
1.7.2.4 Interacción entre HTTP y API REST	17
1.7.2.5 PHYTON.....	17
1.7.2.6 FLASK	17
1.7.2.7 XAMPP	17
1.7.2.8 MySQL.....	17
1.7.3. Elementos de Software para el desarrollo de la carcasa 3D	18
1.7.3.1 FreeCad	18

1.7.4	Elementos de Software para preprocesamiento de datos.....	18
1.7.4.1	Matlab	18
2.	METODOLOGÍA.....	19
2.1	Requerimientos del dispositivo.....	19
2.2	Diagrama de bloques del sistema de medición	20
2.3	Arquitectura del sistema.....	21
2.4	Calibración del sensor.....	22
2.5	Diseño de Hardware	23
2.5.1	Asignación de entradas y salidas.....	23
2.5.2	Circuito de potencia y control.....	25
2.5.2.1	Potencia	25
2.5.2.2	Control.....	26
2.6	Diseño del Software:	28
2.6.1	Software del microcontrolador	28
2.6.2	Software utilizado para el envío de datos en tiempo real.	38
2.6.3	Software utilizado para el preprocesamiento de la información.	40
2.7	Diseño de carcasa 3D	40
3.	Resultados, conclusiones y recomendaciones	44
3.1	Resultados	44
3.2	Conclusiones	55
3.3	Recomendaciones	56
4.	REFERENCIAS BIBLIOGRÁFICAS.....	57
5.	ANEXOS	59
	ANEXO I.....	59
	ANEXO II.....	59
	ANEXO III	60

RESUMEN

El monitoreo de presión ambulatoria ha adquirido gran importancia en la práctica médica convirtiéndose en el primer método para detectar hipertensión arterial y es un eficiente mecanismo de análisis de ciertos tratamientos farmacológicos. La estimación de presión arterial sistólica y diastólica con el método de oscilometría ha permitido desarrollar dispositivos digitales no invasivos para el monitoreo ambulatorio. Bajo este contexto, en el presente trabajo, utilizando el método oscilo métrico, se implementa el software necesario para la automatización del sistema de control de un tensiómetro, se programa la temporización de la automatización que obedece a parámetros recogidos del usuario haciendo uso de una interfaz gráfica sencilla y útil. Los dispositivos utilizados para en monitoreo ambulatorio se caracterizan por acompañar al usuario en sus actividades diarias, por tal razón el prototipo utiliza comunicación WIFI y empleando los protocolos HTTP y Web Socket para el envío y recepción de información el prototipo se convierte en un dispositivo móvil, ya que el procesamiento de los datos y almacenamiento no son realizados en el dispositivo se convierte en indispensable la implementación de una API REST, utilizada como medio de intercambio de información entre varios programas y la base de datos, haciendo uso de la misma se realiza un pre procesamiento de los datos en Matlab para estimar la presión sistólica, diastólica y ritmo cardiaco.

PALABRAS CLAVE: Monitoreo, ambulatorio, Api Rest, parámetro.

ABSTRACT

Ambulatory pressure monitoring has acquired great importance in medical practice, becoming the first method to detect arterial hypertension and is an efficient mechanism for analyzing certain pharmacological treatments. The estimation of systolic and diastolic blood pressure with the oscillometry method has allowed the development of non-invasive digital devices for ambulatory monitoring. Under this context, in the present work, using the oscillometric method, the necessary software for the automation of the control system of a tensiometer is implemented, the timing of the automation is programmed that obeys parameters collected from the user using an interface Simple and useful graph. The devices used for ambulatory monitoring are characterized by accompanying the user in their daily activities, for this reason the prototype uses WIFI communication and using the HTTP and Web Socket protocols for sending and receiving information, the prototype becomes a mobile device, Since data processing and storage are not performed on the device, it becomes essential to implement a REST API, used as a means of exchanging information between various programs and the database, making use of it to perform a data pre-processing in Matlab to estimate systolic and diastolic pressure and heart rate.

KEY WORDS: Monitoring, ambulatory, Api Rest, parameter

1. INTRODUCCION

La presión arterial representa una de las variables fisiológicas más importantes para el personal médico, el registro de presión arterial sistólica y diastólica permite diagnósticos certeros de varias enfermedades de órganos blancos además de brinda información de las respuestas de un paciente a tratamientos farmacológicos, sin embargo sus cifras estan sometidas a errores fenómenos como el de “bata blanca” ha llevado a expertos a buscar nuevas alternativas para registrar valores de presión, cobrando importancia un mapeo de presión arterial durante las actividades diarias del paciente. [1]

La eficacia del mapeo de PA es evidente por sus resultados, permitiendo diagnosticar hipertensión arterial alta enmascarada, dar guía al suministro de fármacos para controla Hipertensión resistente, diagnóstico de hipertensión nocturna, etc. Gracias al método por oscilometría que permite la estimación de presión arterial sistólica y diastólica se han desarrollado dispositivos electrónicos digitales no invasivos para monitorear la PA.

Con estos antecedentes, en el presente trabajo en un primer prototipo, se desarrollará el control para un tensiómetro, dicho control permitirá el envío de datos en tiempo real, para cerciorarse que los datos estan a disposición de otros programas para procesamiento de informacion se incorpora un sistema de recepción y envío de datos el que permita conectarse y administrar la base de datos donde se guardan los registros de presión. Con un preprocesamiento de los datos en tiempo real se verificará la eficiencia de la medición de muestras del dispositivo.

1.1 Objetivo General

Desarrollar e Implementar un sistema de control que permita automatizar el método de adquisición y envío de datos del sensor de presión, además permitirá registrar valores de presión a voluntad del usuario.

1.2 Objetivos específicos.

1. Realizar la revisión bibliográfica de la automatización de toma de datos de presión utilizada para el monitoreo ambulatorio de presión.

2. Realizar revisión bibliográfica de la estructura de equipos no invasivos utilizados para medir presión arterial.
3. Establecer los parámetros configurables utilizados en la programación que llevaran a cabo el proceso de automatización de recolección datos de presión.
4. Identificar requerimientos de hardware que permitan al usuario manipular los parámetros programables para personalizar el sistema acorde a las necesidades de cada paciente.
5. Realizar y ejecutar códigos de programación para: manipular los parámetros configurables por el usuario, control de intervalos de tiempo para toma la presión y envío de datos de presión para su posterior procesamiento.
6. Verificar que el sistema de control implementado adquiera datos del sensor y envíe los datos de presión.

1.3 Alcances

Fase teórica

1. Búsqueda bibliográfica sobre tecnologías y especificaciones de software y hardware de tensiómetros con la finalidad de determinar las características el mismo.
2. Síntesis bibliográfica de la importancia y antecedentes del monitoreo ambulatorio de presión arterial MAPA.
3. Breve revisión bibliográfica de la telemedicina con énfasis de la guía del médico para establecer parámetros en el monitoreo ambulatorio.
4. Estudio y selección de equipo de hardware necesario que permita tomar valores de presión, procesamiento de información, envío de datos para su posterior procesamiento y almacenamiento de datos de presión.
5. Establecer criterios programables de configuración con el objetivo de realizar el monitoreo de presión de 1 a 24 horas.
6. Establecer criterios para determinar un error en el proceso de toma de muestras.

Fase de diseño

1. Se diseñará, en un primer prototipo, una placa electrónica que permita unificar el sistema de control para la adquisición de datos de presión, microprocesador y interfaz con el usuario.
2. Diseño de la carcasa para el equipo en el cual contendrá el sistema de

alimentación, adquisición de datos, microprocesador he interfaz.

3. Se realizará la programación del microcontrolador, en un primer prototipo, que permita: configuración de parámetros requeridos para automatizar el método, ejecutar la configuración y adquirir la presión.
4. Realizar la programación necesaria para comandar un interfaz HMI, permitirá visualizar al usuario las configuraciones y los valores de presión sistólica y diastólica los cuales serán almacenados.

Fase de implementación

1. Implementar la placa PCB, en un primer prototipo, que unifica los sistemas de sistema de control he interfaz con el usuario.
2. Implementación de una pantalla grafica que muestre el valor de presión y botones para la configuración del sistema.
3. Implementar los códigos de programación necesarios para realizar el sistema de control que permitirá tomar muestras a intervalos de tiempo programado por el usuario a través de pulsadores.

1.4 Marco teórico

1.4.1 Presión Arterial

La principal función del sistema cardiovascular es el transporte y distribución de sangre en un sistema cerrado formado por el conjunto de vasos sanguíneos y corazón, el corazón representa el núcleo del sistema cardiovascular, sus paredes conformadas de musculo cardiaco que reacciona a estimulo eléctrico comprimen y contraen al corazón para bombear la sangre a todos los vasos sanguíneos, como resultado de la acción de bombeo del corazón, la sangre, que fluye a través venas, arterias y capilares, se encuentra bajo presión.

Por tanto, la presión de sistema cardiovascular simboliza la presión que la sangre ejerce sobre las paredes de los vasos sanguíneos, la presión arterial produce tres tipos de energía que son elástica, gravitacional y cinética. La energía elástica producida en las paredes que se estiran cuando pasa un determinado volumen de sangre y en función de las propiedades elásticas crea la presión. La relación entre el volumen y presión no es constante pues las estructuras vasculares no son homogéneas, por la naturaleza cíclica del llenado y vaciado

cardiaco el flujo es pulsátil por tanto existirán cambios cíclicos en la presión arterial, la acción de la contracción ventricular produce la mayor presión denominada presión sistólica y la presión menor denomina presión diastólica se da cuando el corazón está en reposo o fase de diástole, que representa a la presión que ejerce la sangre sobre las arterias entre dos latidos

Los valores normales a nivel grandes arterias como la humeral corresponde a 120 mmHg como presión sistólica y 80 mmHg como presión diastólica, sin embargo, este valor corresponde a un ideal y los niveles de presión están determinados por el sexo y edad en la siguiente Tabla 1.1 muestra los niveles de presión arterial normales. [1]

Tabla 1.1 Rangos normales de presión arterial [2]

Edad (años)	Presión Sistólica		Presión Diastólica	
	Hombre	Mujer	Hombre	Mujer
21 – 25	120.5	115.5	60 – 86	70.5
26 – 30	119.5	113.5	62 – 88	71.5
31 – 35	114.5	110.5	65 – 89	72.5
36 – 40	120.5	112.5	68 – 92	74.5
41 – 45	115.5	116.5	70 – 96	73.5
46 – 50	119.5	124	70 – 98	78.5
51 – 55	125.5	122.55	70 – 100	74.5

Valores de presión fuera de los rangos normales representan enfermedades cardiovasculares, enfermedades renales o diabetes. Pacientes diagnosticados con presión arterial alta o hipertensión se caracterizan por tener problemas con el flujo sanguíneo enviado desde el corazón o problemas con la resistencia elásticas de las paredes arteriales. Con paredes poco distendidas y a un flujo alto de sangre normal se incrementará la presión arterial, pacientes sin tratamiento y con prolongada presión arterial elevada pueden sufrir ataques cardiacos o accidentes cerebro vasculares.

Una presión arterial alta, o también llamada hipertensión secundaria, puede ser causada por diversas enfermedades, estos trastornos pueden ser: enfermedades renales, problemas con la glándula tiroides, problemas congénitos en los vasos sanguíneos, tumores a nivel de la glándula suprarrenal o consumo de drogas ilícitas

Una herramienta importante para el diagnóstico, control y tratamiento en la hipertensión es la monitorización ambulatoria de presión arterial (MAPA) [3]

1.4.2 Medición automatizada de la presión

El proceso de automatización para la medición de presión se denomina en la práctica médica como monitorización ambulatoria de presión arterial conocida por sus siglas como MAPA la cual ha llegado a convertirse en un importante hallado para el diagnóstico de enfermedades por la certeza de sus cifras, incluso se menciona como el primer diagnóstico para confirmar elevadas cifras de presión arterial.

El monitoreo ambulatorio de presión arterial consiste en monitorear la presión arterial durante 24 horas el estudio presenta una visión de tensión arterial y ritmo cardiaco, la importancia de monitorear al paciente en su vida diaria radica en que existe un error en la medición de tensión arterial en los consultorios, los valores suelen presentarse elevados lo que se llama fenómeno de “bata blanca”, el mapeo de presión arterial muestra variaciones de cifras de presión arterial tomados: en consulta, mediciones en la vida cotidiana y durante el sueño.

Las cifras arrojadas por MAPA permiten clasificar a los pacientes en dos grupos, pacientes con hipertensión arterial y pacientes con hipertensión oculta, en un paciente “falso controlado” las cifras detectan momentos del día con presión arterial se eleva.

Una presión arterial alta recogida en un MAPA de 24 horas esta correlacionada con un daño de un órgano blanco, por tanto, MAPA predice la mortalidad cardiovascular.

Una ventaja de MAPA es el único método que mide la presión arterial nocturna, varios estudios muestran que la presión arterial nocturna permite tener mejores pronósticos, razón por la cual los médicos recomiendan realizar mediciones las 24 horas por varias razones como actividades físicas y mentales que afectan a la estandarización de mediciones de presión arterial diurnas. [4]

1.4.3 Equipo para monitorización ambulatoria de presión.

La variación de las cifras tomadas en consulta en comparación con las tomadas en el domicilio del paciente añadido a esto la necesidad del médico por analizar la evolución de un paciente con hipertensión arterial llevo a varios científicos a desarrollar un equipo con el objetivo de realizar una monitorización ambulatoria de presión arterial, los científicos liderados por Sir George Pickering quien afirmaba en 1972 que “la relación entra presión arterial y mortalidad es cuantitativa, cuanto más alta es la presión peor es el diagnostico”, desarrollo un equipo con una cánula intraarterial, los avances tecnológicos a partir de esa

época han permitido encontrar métodos no invasivos y más prácticos para la toma de presión arterial.

El monitoreo constante de PA se logró gracias al descubrimiento del método oscilo métrico, utilizado por primera vez en 1960. Los datos recogidos del descubrimiento eran satisfactorios pues se evidencio que la presión arterial tomada en el domicilio del paciente es menor por 10 mmHg. Demostrando que un supuesto monitoreo en la consulta médica no era un método asertivo para el pronóstico de un paciente hipertenso. [5]

Los tensiómetros que utilizan el método oscilométrico, han ido mejorando y en la actualidad gracias al avance de la tecnología los dispositivos son muy livianos y pequeños, conformados por una almohadilla oclusora cuyo mecanismo de inflado dependerá de una bomba de aire y una válvula. Los modelos comerciales digitales presentan tensiómetros de brazo y de muñeca. En el presente trabajo se desarrollará un tensiómetro de muñeca del cual se detallará a continuación los elementos constituyentes.

1.4.3.1 Diseño de hardware general de un tensiómetro digital de muñeca.

Los tensiómetros automáticos están a merced de una misma estructura mecánica, compuestos por un brazalete o manguito de presión el cual se infla de manera automática con la ayuda de una bomba de aire, electroválvula, microcontrolador y sistema de energía [6].

La Figura 1.1 se visualiza los elementos que conforman el tensiómetro.

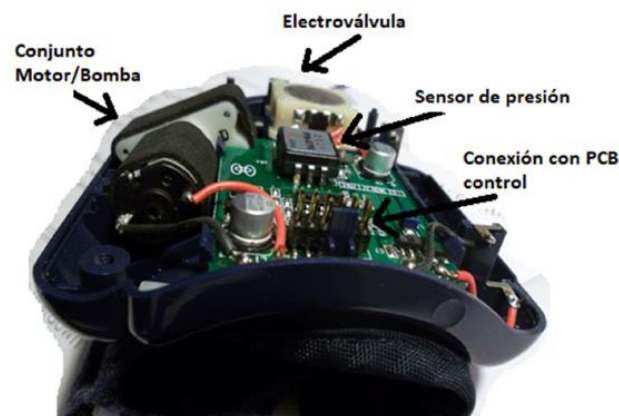


Figura 1.1. Elementos que conforman un tensiómetro digital [6]

Microcontrolador: El microcontrolador es el cerebro y tendrá la capacidad para controlar la toma de muestras de presión, control de bomba de aire, control de válvula, admitir periféricos como pantalla y sensores.

Sistema neumático: Conformado por

- 1) La almohadilla oclusora: Utilizada como depósito de aire para ocluir la arteria humeral.
- 2) La bomba neumática: Responsable de generar la diferencia de presión y el caudal necesario para la presurización instantánea necesaria en la almohadilla.
- 3) El controlador de presión llevado a cabo por electroválvulas que manejan la conmutación del segmento neumático de la almohadilla.

El circuito electrónico de potencia responsable de energizar y regular el ciclo de trabajo de la bomba y las electroválvulas.

Sensor de presión: Los sensores de presión dirigidos a aplicaciones médicas tales como tensiómetros digitales capaces de tomar datos en lapsos de tiempo, cifras que son transmitidos hacia una interfaz para ser evaluadas.

Una visión más detallada acorde a la funcionalidad de cada elemento permite clasificarlos por sistemas por tanto se tendrá un sistema de control, interfaz y sensor que junto con el brazalete tomaran datos de la variación de presión que se someterá la muñeca. Como se puede ver en la Figura 1.2. [7]

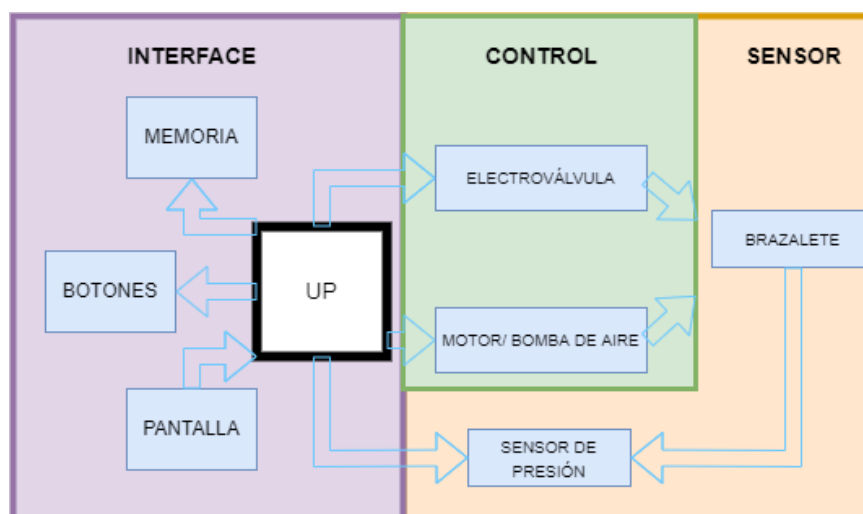


Figura 1.2. Sistemas que conforman un tensiómetro

1.4.3.2 Automatización del sistema de control para medición de presión.

Es necesario que se automatice el proceso para la medición de presión. El microcontrolador, utilizando señales eléctricas se encargará de realizar el control neumático, el diagrama PID de la Figura 1.3 describe el lazo cerrado del proceso [8]

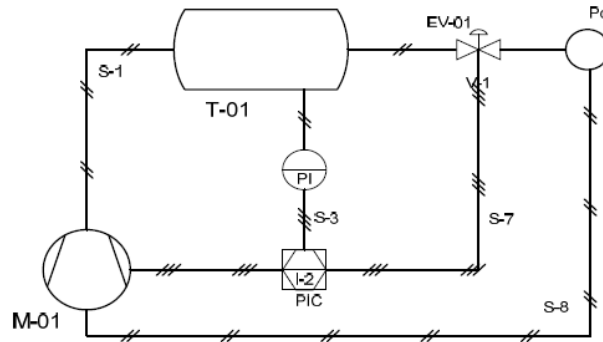


Figura 1.3. Diagrama PID [8]

Donde:

P_0 : Aire a presión atmosférica.

T -01: Brazaleta Oclusivo.

EV 01: Electroválvula 2 posiciones.

PIC: Controlador indicador de presión

PI: Indicador de presión.

M 01: Bomba

El lazo inicia con el ingreso de aire a presión atmosférica ayudado por la bomba de aire, comandada por el sistema microprocesador, el flujo controlado ingresará al depósito o brazaleta oclusivo, el controlador recibirá señales del indicador de presión y de acuerdo con un set point programado envía señales a la electroválvula para encenderse o apagarse. [8]

1.4.3.3 Métodos de calibración para sensores de presión con salida eléctrica.

El sensor es el elemento encargado de captar las variaciones de presión del medio y transforma en variaciones de voltaje o variaciones de otra unidad.

Los transmisores y transductores de presión con salida eléctrica están generalmente en el orden de los mA o mV.

La relación entre la señal eléctrica y presión media para transmisores y transductores de presión es:

$$P = \frac{R_p}{I_E - I_E} (I - I_E) \quad (1.1)$$

Donde

- R_p es el rango de medida en unidades de presión
- I y I_e son los valores inferior y superior de rango eléctrico de instrumento.

Para una adecuada calibración es necesario

1. Patrones: Se utiliza uno o varios patrones de presión, los cuales tendrán como incertidumbre máxima del patrón es la cuarta parte de la esperada de la calibración.
2. Generador y controlador de presión.

Es necesario un equipo que genere la presión de forma lenta y que pueda mantener esta presión de forma estable. Lo idóneo sería que el generador de presión tenga una mejor resolución que el patrón

3. Condiciones ambientales

Se sugiere mantenerse dentro de rangos de temperatura ambiente entre 3 y 20°C y una humedad relativa menos a 60%

4. Accesorios

Es necesario la utilización de tubos y racores para controlar el fluido. En caso de tener un transductor es necesario una fuente de energía apropiada. [9]

Pasos para una calibración:

1. Comprobación inicial, se analizará un ciclo de medidas, las cifras acompañadas con un cálculo de incertidumbres serán un indicativo del estado del sensor y si es necesario su ajuste
2. Determinar si es necesario el ajuste
3. Calibración

1.4.3.4 Método de estimación de presión arterial en dispositivos digitales.

Oscilometría, es el principal método utilizado por dispositivos digitales para la medición de presión arterial, a partir de métodos mecánicos se crea un oscilograma con las amplitudes de oscilaciones del volumen sanguíneo correspondientes a la presión externa, la variación en amplitud de las oscilaciones permite estimar la presión arterial, existen tres algoritmos más populares previamente estudiados y corroborados.

Algoritmo de la máxima amplitud: El algoritmo revela que a la mayor amplitud de una oscilometría estima la presión arterial media y es igual a la presión externa ejercida, sus ecuaciones son:

$$P_{max} = \frac{\alpha}{\alpha+\beta} P_S + \frac{\beta}{\alpha+\beta} P_d \quad (1.2)$$

$$\alpha = 11.4 \pm 0.9 \text{ mmHg} \quad (1.3)$$

$$\beta = 13.8 \pm 0.7 \text{ mmHg} \quad (1.4)$$

Donde:

P_{max} = Presión máxima

P_e = Presión externa

Alfa y Beta representan los regímenes de presión transmural positiva y negativa de la curva de distensibilidad arterial.

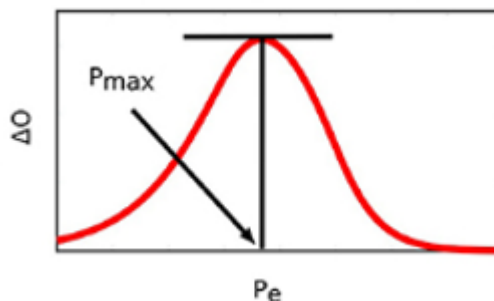


Figura 1.4. Curva de máxima amplitud [10]

Algoritmo derivativo: Se estima presión sistólica y diastólica a partir de la máxima y mínima pendiente del oscilograma y es igual a la presión externa, sus ecuaciones son:

$$P_{minslope} = P_S + \frac{PP}{e^{\alpha} + 1} \quad (1.5)$$

$$P_{maxslope} = P_d + \frac{PP}{e^{\frac{PP}{\beta}} + 1} \quad (1.6)$$

$$\frac{PP}{e^{\frac{PP}{\alpha}} + 1} = 0.9 \pm 0.3 \text{ mmHg} \quad (1.7)$$

$$\frac{PP}{e^{\frac{PP}{\beta}} + 1} = 1.4 \pm 0.4 \text{ mmHg} \quad (1.8)$$

Donde:

PP= Presión sistólica – Presión Diastólica

P_e = Presión externa

Alfa y Beta ya definidos en el primer método.

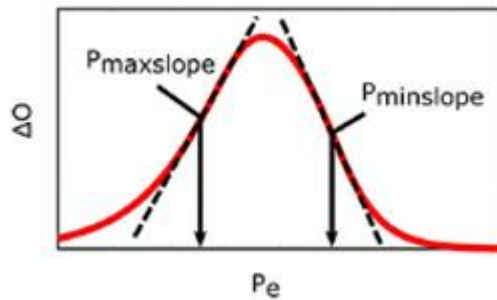


Figura 1.5. Estimación por algoritmo derivativo [10]

Algoritmo de razón fija: A partir de una porción ascendente y descendente del oscilograma se fija una relación del valor máximo, se fija fórmulas para las razones verdaderas para determinar las amplitudes en el oscilograma que son estimaciones de presión sistólica y diastólica [10]

$$TR_S = \frac{2\alpha - (PP + 2\alpha)e^{-\frac{PP}{\alpha}}}{(2(\alpha + \beta) - (PP + 2(\alpha + \beta)))e^{-\frac{PP}{\alpha + \beta}}} \quad (1.9)$$

$$TR_d = \frac{2\beta - (PP + 2\beta)e^{-\frac{PP}{\beta}}}{(2(\alpha + \beta) - (PP + 2(\alpha + \beta)))e^{-\frac{PP}{\alpha + \beta}}} \quad (1.10)$$

$$PR_S = P_S - [2(\beta + \alpha) - (PP + 2(\beta + \alpha))e^{-\frac{PP}{\beta + \alpha}}](R_S - T_{RS}) \quad (1.11)$$

$$PR_d = P_d - [2(\beta + \alpha) - (PP + 2(\beta + \alpha))e^{-\frac{PP}{\beta + \alpha}}](R_d - T_{Rd}) \quad (1.12)$$

Donde:

TRs: Amplitud del oscilograma que corresponde a la presión arterial sistólica real.

TRd: Amplitud del oscilograma que corresponde a la presión arterial diastólica real

PP, Alfa y Beta ya fueron definidos en los métodos 1 y 2.

PR_s : Presión sistólica real

PR_d : Presión diastólica real

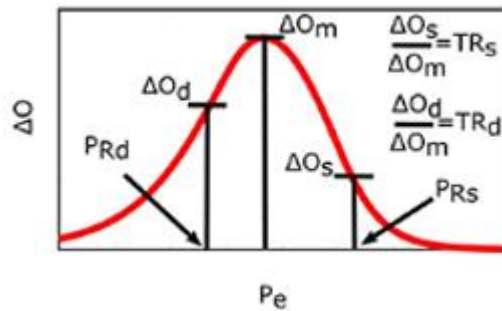








Figura 1.6. Estimación por algoritmo de razón fija [10]

1.5 Elementos de Hardware utilizados para el sistema de control e Interfaz

Tabla 1.2 Elementos utilizados para el sistema de control e Interfaz

Elemento	Características
<p>ESP32</p> 	<p>Procesador</p> <ul style="list-style-type: none"> - CPU de 32 bits con una velocidad de 240 MHz de simple núcleo con 630 DMIPS. - Tiene un coprocesador ULP de ultra baja energía. <p>Memoria:</p> <ul style="list-style-type: none"> - SRAM de 520 KB - Mbyte de memoria flash dentro del modulo <p>Conexión inalámbrica:</p> <ul style="list-style-type: none"> - Tiene un transceptor Wi-Fi 802.11 b/g/n - Bluetooth de modo dual integrado


	<p>Conexión con periféricos:</p> <ul style="list-style-type: none"> - 2 x SPI - 2 x interfaces para I2C - 2 x interfaces para I2CS - 2 x UART - Cada GPIO tiene salida y salida temporizada para PWM - Conexión esclavo – maestro SDIO de 50 Mhz.
<p>Electroválvula KSV05</p> 	<ul style="list-style-type: none"> - - Voltaje nominal: 3 V – DC - Corriente nominal: 75 mA - Resistencia: 40Ω +- 10% - Aplicaciones: Aire
<p>Bomba de aire KMP14A</p> 	<ul style="list-style-type: none"> - Voltaje max: 6V - Voltaje nominal: 3V - Corriente maxima:380 mA - Presión de flujo max: 350 mmHg.
<p>Sensor de presión MPS20N0040D</p> 	<ul style="list-style-type: none"> - Voltaje de alimentación: 3.3 V a 5 V - Rango de presión: 0 a 5.8 PSI (0- 300 mmHg). - Linealidad: ± 0.3% F.S. - Histéresis: ± 0.7% F.S. - Voltaje Bias: ± 25 mV. - Voltaje máximo a fondo de escala: 50 a 100 mV. - Salida del módulo: Digital incorpora un conversor AD de 24 bits. - Amplificador con ganancia de 128. - Temperatura de operación: -40 a 80 C - Regulador de fuente de alimentación para sensor y chip

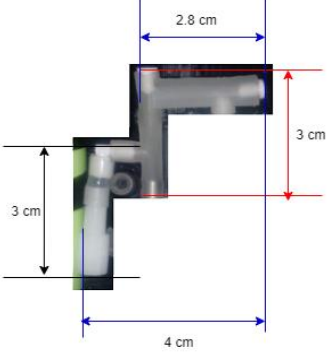

<p>Pantalla OLED</p> 	<ul style="list-style-type: none"> - Modelo: SH1106 - Tamaño: 1.3" - Display: Azul 128 x 64 - Voltaje de alimentación: 3 – 5 v - Interfaz de comunicación: I2C - Ultra bajo consumo de energía. - Peso: 50 g
<p>Pulsadores</p> 	<p>Pulsadores de dos pines que cierran al ser oprimidos, se utilizaran como entradas al microcontrolador</p>

1.6 Elementos de hardware utilizados para el sistema neumático

Para el sistema neumático se utiliza una bomba y una electroválvula para el ingreso y salida de aire que ya fueron descritos en el sistema de control.

Tabla 1.3 Elementos utilizados para el sistema neumático.

Elemento	Características
<p>Tensiómetro digital Homedics</p> 	<p>Tensiómetro digital Modelo MPW- O200 de HoMedics, cumple con requerimientos de prueba de presión con ciclos simulados de acuerdo con EN 1060-3.</p>
<p>Tubería neumática</p>	<p>Se utilizo el sistema de tubería proveniente del tensiómetro comercial Homedics Conecta a la bomba, válvula y sensor para la transferencia eficiente de aire</p>

	
<p>Almohadilla Oclusiva</p> 	<p>Depósito de aire conectado directamente al conductor neumático trasfiere el aire de la bomba y válvula, realiza la oclusión de la arteria radial. Compuesto por tres elementos.</p> <ul style="list-style-type: none"> - Bolsa inflable encargado de comprimir la muñeca - Carcasa deformable que se fija a la carcasa del dispositivo por medio de agarraderas. - Envoltura de tela

1.7 Elementos de Software

Para cubrir dos alcances del presente proyecto uno de los cuales son envío, recepción de datos para su análisis, procesamiento y almacenamiento, el segundo es el diseño de una carcasa que contenga el sistema de energía y control se utilizaran las siguientes herramientas.

Por último, se describirá el software utilizado para un preprocesamiento de la información.

1.7.1 Elementos de Software para programar ESP32

1.7.1.1 Plataforma de desarrollo Arduino

Entorno de desarrollo de código abierto, multiplataforma y gratuito, provee de un editor de texto dinámico y fácil de usar conocido como IDE utilizado para crear el firmware y cargar el programa a placas compatibles, su lenguaje de programación está basado en C y C++.

Espressif empresa de desarrollo de chips ESP32, faculta la compatibilidad con dos IDEs PlataformaIO e IDE de Arduino. [11]

1.7.2 Elementos de Software para el envío y recepción de datos

1.7.2.1 Formato tipo JSON

Formato preferido para el intercambio de datos, a través de la Webs, su sintaxis está basada en el lenguaje JavaScript, sin embargo, cualquier lenguaje de programación puede consumir datos en formato tipo JSON por su sencillez y facilidad de utilización. Emplea arrays y objetos como elementos de sintaxis.

Arrays: Corresponden a matrices de valores: numéricos, cadenas, objetos o matrices.

Objetos: Permite asignar valores en parejas denominadas nombre/valor los cuales estan separados por dos puntos.

```
{"time_start": "14/09/2018 11:59:07", "temp_001": 25, "Country": "Ecuador", "City": "Guayaquil", "description": "Cool"},
```

El formato ha tomado gran popularidad entre los desarrolladores pues brinda un lenguaje estandarizado para los documentos, establece un protocolo preciso para enviar solicitudes y respuestas en una interfaz de comunicación API utilizando el protocolo HTTP. [12]

1.7.2.2 Protocolo de transferencia HTTP

El protocolo de siglas HTTP por su nombre Hypertext Transfer Protocol, protocolo cliente – servidor gestiona la operación de solicitud respuesta, el cliente envía una petición de datos y el servidor responde la solicitud.

Utilizando el modelo TCP/ IP, realiza la navegación para el envío de solicitudes de los clientes Web, mantiene la conexión para asegurar el libre intercambio de datos.

El protocolo HTTP tiene comandos con los cuales establece un dialogo con el servidor, los más utilizados son:

GET: Con el cual recoge informacion desde un servidor, HTTP envía la informacion de una URL

POST: Envía informacion al servidor utiliza, comúnmente, un formulario HTTP, los datos elegidos se envían en una cadena de texto [13]

1.7.2.3 API REST

Es un servicio Web que está diseñado para lograr intercambio de informacion entre dos servicios Web, está conformado por API la cual define las reglas para conseguir comunicarse con otros sistemas de software, la API provee de recursos y funciones para facilitar la interacción e intercambio de informacion entre programas del computador. Rest

proporciona a la API la arquitectura para la funcionalidad, en conjunto las API basadas en la arquitectura REST logran comunicación entre usuarios que manejen HTTP, la funcionalidad es permitir que otro software acceda a la información enviada. [14]

1.7.2.4 Interacción entre HTTP y API REST

Una API REST cubre el protocolo de transferencia HTTP, por medio de especificaciones de interacción que vienen desde HTTP como son los verbos GET, DELETE, POST, PUT, etc. Los más importantes y utilizados en esta aplicación ya fueron descritos. [14]

1.7.2.5 PYTHON

Lenguaje de programación de tipado dinámico orientado a objetos que se caracteriza por ser un lenguaje interpretado ya que se ejecuta utilizando un programa intermedio convirtiéndolo en un lenguaje flexible y portable, su tipado dinámico evita la declaración de tipo de dato determinado durante la ejecución,

El lenguaje tiene una sintaxis clara y simple por su gran cantidad de librerías, su intérprete está disponible en varias plataformas como Windows, Linux Mac, etc. [15]

1.7.2.6 FLASK

Flask es un pequeño framework, utilizado en Python, brinda un núcleo sólido para servicios básicos y para servicios más complejos utiliza extensiones, que se pueden escoger y descargar, con estas funcionalidades permite desarrollar todo tipo de aplicaciones con Python. [16]

1.7.2.7 XAMPP

Programa desarrollado por Apache Friends, permite una rápida instalación de una pila WAMP o LAMP de manera fácil y rápida. Su nombre corresponde a un acrónimo de los programas que los componen. Los cuales solo se procederá a definir los utilizados en el presente trabajo. [17]

1. Apache: Es un servidor WEB de código abierto multiplataforma.
2. MySQL: Junto con el servidor WEB Apache, lenguaje PHP y MySQL logra almacenar datos que se utilizarán para servicios WEB

1.7.2.8 MySQL

Base de datos de código abierto la cual organiza datos empleando tablas, maneja el lenguaje SQL para modificar, crear y extraer datos, está basado en un modelo cliente servidor, los clientes que necesiten, se conectarán al servidor que gestiona la base de datos realizando una solicitud de información, el servidor regresará la ejecución de la

solicitud cuando tenga derechos de acceso. Al utilizar XAMPP el lenguaje de programación del servidor será PHP.

El lenguaje SQL esta orientado para manejar bases de datos, por medio de sentencias realiza operaciones y manipulación de informacion de una base de datos, las cuales se ejecutan con el comando SELECT las sentencias pueden ser FROM, WHERE y ORDER BY [18]

1.7.3. Elementos de Software para el desarrollo de la carcasa 3D

1.7.3.1 FreeCad

Programa de software libre y gratuito para el diseño 3D asistido por computadora, permite un modelado basado en GUI y modelado basado en Scripts utilizado el lenguaje de programación Python, su código está basado fundamentalmente por la librería OpenCasCade que contiene un poderoso núcleo para modelar en 3D

Sus dos opciones de modelado permiten al programa ser utilizado por diseñadores gráficos y por programadores. [19]

1.7.4 Elementos de Software para preprocesamiento de datos.

1.7.4.1 Matlab

Nombre proveniente de Matrix LABoratory, es un lenguaje diseñado para computación, técnica matemática, modelado, análisis, procesamiento de datos, simulación y representación de gráficos, incorpora librerías llamadas toolboxes que se centran en resolver problemas específicos

App Designer: Entorno de desarrollo para diseñar aplicaciones, integra el potente editor de MATLAB y el componente IU para interfaces interactivas, se puede realizar distribución de la aplicación empaquetándola. [20]

2. METODOLOGÍA

El presente capítulo se expone las necesidades del equipo, identificando los requerimientos del dispositivo. Dichas necesidades determinaran los elementos y mecanismo de funcionamiento para realizar el control del dispositivo.

2.1 Requerimientos del dispositivo

Tabla 2.1 Requerimientos del dispositivo

Requerimiento	Objetivo para implementar
Configuración: Realizar un monitoreo ambulatorio de presión durante 24 horas, aceptando configuraciones del usuario como: periodo de estudio variable de 1 a 24 horas, periodo sueño y frecuencia de las muestras que son los lapsos de tiempo entre cada muestra	<ol style="list-style-type: none">1. Utilizar un sistema de que provea energía durante 24 horas.2. Implementación de un teclado que permita al software aceptar configuraciones del usuario3. Control de la adquisición de presión.4. Implementación y control del ingreso del aire controlando la bomba y la válvula.5. Ingreso en modo ahorro de energía entre cada muestra.6. Incorporación de interfaz gráfica.
Monitoreo de tiempo real de cifras de presión.	<ol style="list-style-type: none">1. Envió en tiempo real los datos vía WIFI utilizando el protocolo HTTP.
Almacenamiento de la información de presión.	<ol style="list-style-type: none">1. Implementación de una API REST como mecanismo de recepción y envío de datos en tiempo real, la interfaz permite el consumo de datos a otros programas2. Utilización de MySQL, base de datos que almacenara los datos de presión.

Equipo de fácil manejo, la programación de los parámetros debe ser intuitiva	Programación de un menú intuitivo que permita al usuario ingresar los parámetros.
El equipo debe ser portátil para poder realizar el monitoreo de presión de 24 horas, evitando que el equipo intervenga con la movilidad del usuario.	Diseño de una carcasa ligera y ergonómica que incorpore los sistemas de: control, interfaz, sensor y energía.

2.2 Diagrama de bloques del sistema de medición

Para realizar un diseño de hardware y software apropiado, primero se estableció un diagrama claro con los elementos del sistema y componentes utilizados para el envío y preprocesamiento de datos en la Figura 2.1 se observa un diagrama de bloques del sistema.

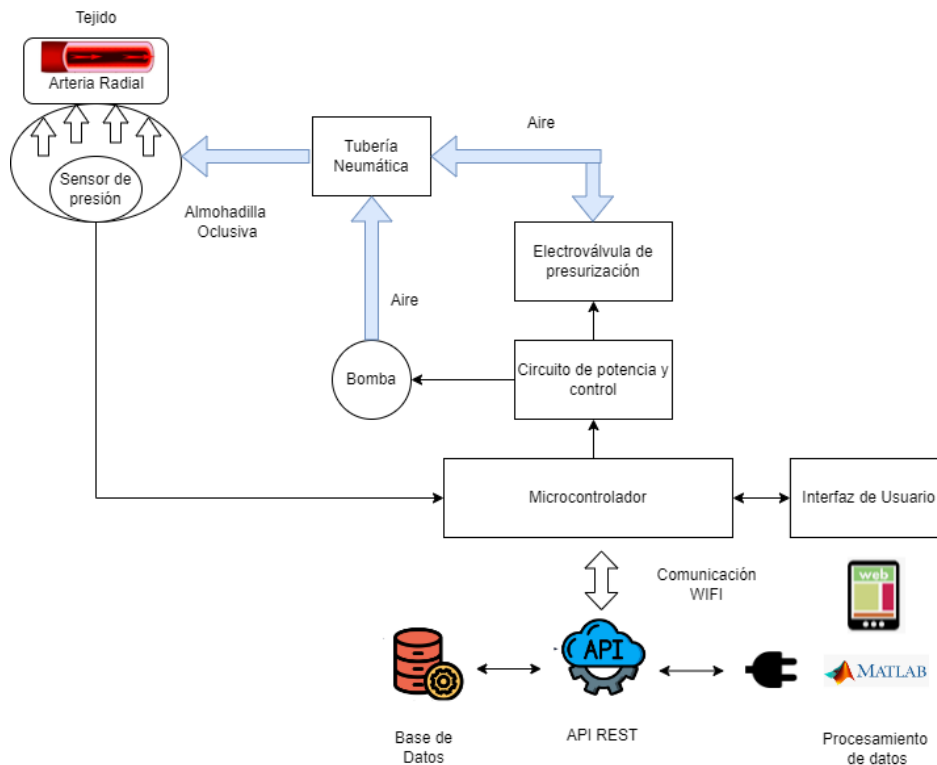


Figura 2.1. Diagrama de bloques del dispositivo.

2.3 Arquitectura del sistema

Acorde a las necesidades se describirá como un dispositivo integrador de varios sistemas capaz de monitorear la presión en tiempo real con conexión inalámbrica, el cual debe ser de fácil manejo, portátil, con un software intuitivo y una carcasa ergonómica evitando interferir con las actividades diarias del usuario.

Para lograr este fin se presenta la siguiente arquitectura para el prototipo en la Figura 2.2.

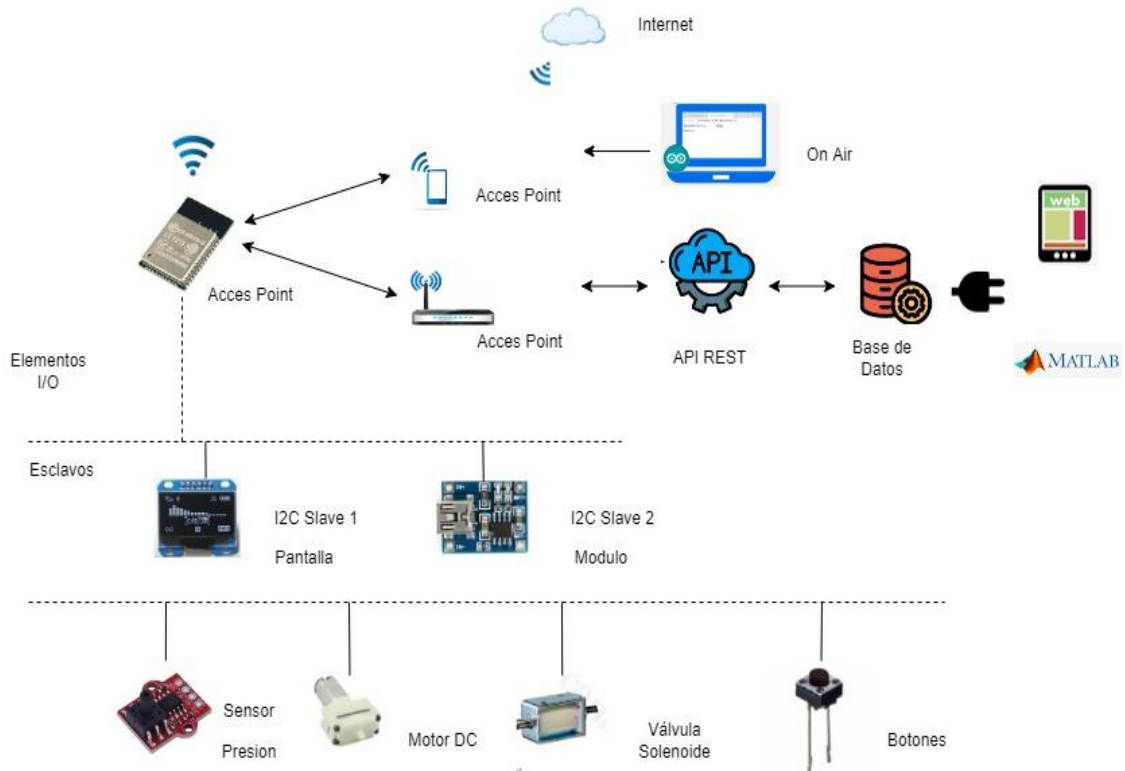


Figura 2.2. Arquitectura del tensiómetro

En el primer nivel por medio de entradas y salidas del microcontrolador se realiza: la adquisición de datos del sensor de presión, utilizando botones se recepta configuraciones del usuario, se controla la válvula solenoide y la bomba con una salida PWM los dos últimos utilizados en conjunto para controlar el ingreso y salida del aire que permitirán al sensor captar la presión dentro de la banda de aire.

Se utiliza comunicación I2C para la recepción de datos del módulo que monitorea la energía y para la pantalla.

Para enviar datos se utilizó WIFI que viene incorporado en el chip ESP32, se dota al dispositivo con autonomía para conectarse a una nueva red WIFI utilizando un teléfono celular, por tanto, es necesario que primero el microcontrolador actúe como un Access

Point, aceptando el acceso a la red a cualquier dispositivo inteligente, el que será encargado de re direccionar la conexión del chip a otro punto de acceso que puede ser un router o un teléfono inteligente esta red realizara la conexión entre la API, por último el prototipo pasa a configuración Station(STA) y Access Point (AP) a la vez. La configuración posibilita al dispositivo acceder a cualquier red con internet utilizando el teléfono celular.

El acceso a una red WLAN permite la conexión de dispositivo con la API, con códigos POST y GET del protocolo HTTP envía y recepta información con el servidor, envió para su procesamiento y recepción para visualizar el resultado del procesamiento de datos en la pantalla.

Para realizar actualización de Software inalámbrico (On Air) desde cualquier dispositivo inteligente, el prototipo mantiene la configuración de Access Point y agregando el protocolo de comunicación WebSocket posibilita que la comunicación entre le cliente y servidor quede abierta, el envío y recepción de datos se mantiene de forma dúplex, es asi que la información es enviada al cliente sin solicitarlo.

Los datos de presión son enviados en formato tipo JSON a una API REST, la cual los almacena en una base de datos, la API permite la conexión a otros programas o páginas Web, de este modo utilizando POST y GET se conecta con Matlab para el preprocesamiento de la información.

2.4 Calibración del sensor

Procedimiento:

1. Se implementará el sistema mecánico como se visualiza en la Figura 2.3, se utilizan racores, tubos neumáticos, tubos de silicona y acopladores neumáticos para evitar fugas.
2. Se tomará valores de presión a la salida del sensor cuando la presión esta estática conectando la salida al Arduino.
3. Se procederá a tomar cifras de presión para dos curvas: curva de subida y curva de bajada.
4. Se procederá a ir desde 0 hasta 210 mmHg con intervalos de 10 mmHg
5. Acorde a los datos se determinará la verdadera sensibilidad de sensor.
6. Obtención de la ecuación de la curva.
7. Determinar el método de linealización de la curva.

8. Determinar la eficiencia de la ecuación comparando presiones obtenidas con tres dispositivos medidores de presión: manómetro y dos sistemas digitales.

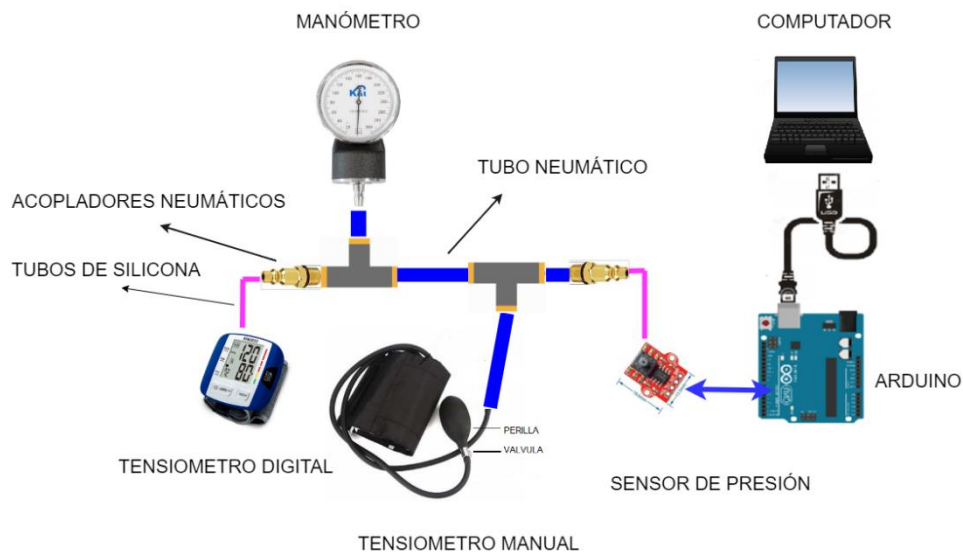


Figura 2.3. Sistema para calibración del sensor

2.5 Diseño de Hardware

2.5.1 Asignación de entradas y salidas.

De acuerdo con los requerimientos del prototipo de la Tabla 2.1 y los elementos seleccionados mencionados en el capítulo 1 el microcontrolador requiere

Tabla 2.2 Requerimientos del chip

Entradas	Salidas
4 botones que funciones como subida, bajada, seleccionar y prender, para aceptar las configuraciones de usuario.	Comunicación I2C para conexión de pantalla y modulo energía
1 entrada para recepción de datos del sensor	Comunicación WIFI
	1 salida para control PWM de Bomba
	1 salida para control de la válvula
	1 salida para activación y envío de señal de reloj al sensor.

Se requiere 5 entradas, 2 salidas PIN/OUT, 1 salida PWM, comunicación I2C y WIFI por tal razón se eligió el microcontrolador ESP32.

En la Figura 2.4 se visualiza la asignación de entradas y salidas del microcontrolador se adjunta la Tabla 2.4 que detallara las funciones de cada PIN

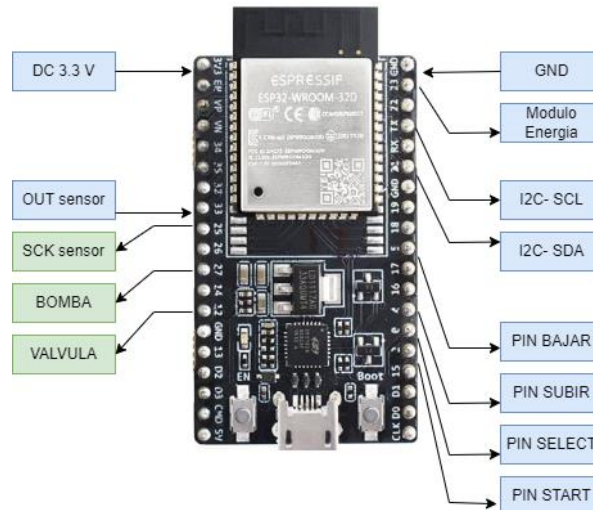


Figura 2.4. Distribución de entradas y salida en la ESP32

Tabla 2.3 Distribución de pines del ESP32

Señal	Tipo de señal	PIN	Función
Alimentación 3.3 V.	Suministro DC - INPUT	1 - 3,3 V	Suministro de 3,3 V proveniente del microsistema de energía
OUT Sensor	Señal digital – Input	8 – GPIO 33	Recepción serial de datos del sensor de presión.
SKC Sensor	Señal digital – Output	9 – GPIO 25	Control de encendido del sensor y envío de señal de reloj
Bomba	Señal digital – Output	11 – GPIO27	Control de la bomba por señal PWM.
Válvula	Señal digital- Output	13 – GPIO14	Control de la válvula encendido y apagado.
GND	Suministro GND – Input	38 – GND	Conexión a GND del microsistema de energía.
Activación de periféricos.	Señal digital – Output	37- GPIO 23	Activa la energía para los dispositivos periféricos.

SCL	Comunicación I2C	36 – GPIO 22	Comunicación con la pantalla y módulo de energía
SDA	Comunicación I2C	33 – GPIO 21	Comunicación con pantalla y módulo de energía.
START - botón	Señal digital – Input pull up	24 – GPIO2	Ingreso de configuraciones.
SELECT- botón	Señal digital – Input pull up Interrupción externa RTC	25 – GPIO0	Ingreso de configuraciones. Pin programado para despertar del modo Deep Sleep.
SUBIR - botón	Señal digital – Input pull up	26 – GPIO4	Ingreso de configuraciones.
BAJAR - botón	Señal digital – Input pull up	29 – GPIO5	Ingreso de configuraciones.

2.5.2 Circuito de potencia y control

2.5.2.1 Potencia

La adquisición y procesamiento de los datos de presión se realiza mediante un microcontrolador, el sistema embebido, con la ayuda del sensor, registra los datos de presión provenientes del interior de la manga de aire cuando esta se desinfla. La adquisición de los datos empieza con el inflado de la manga gracias a la acción de la bomba KPM14A comandada mediante una señal PWM, con rangos máximos de 6V a 380 mA y flujo de presión 350 mmHg como se mencionó en el capítulo 1, la acción de la bomba continua hasta registrar una presión dentro de la manga de 200 mmHg, entonces se procede a desinflar la manga. En el proceso de desinflado se utiliza la electroválvula KSV05.

El microcontrolador ESP32 entrega una corriente típica de 4mA con un valor máximo de 16 mA, por lo que es necesaria una etapa de amplificación de corriente para comandar la bomba, para esta etapa se utilizará un amplificador operacional 2222A en modo emisor común. El transistor se someterá a corte y saturación acorde a la hoja de datos mostrada en el ANEXO 1, alcanzando una corriente de colector en saturación de 150 mA.

Para separar la etapa de control de la etapa de potencia se usó un optoacoplador 4N25, sus características eléctricas el ANEXO 2.

Datos útiles del diodo (optoacoplador) :

Forward Voltaje Máximo: 1.5 V

Se diseñará para una corriente de 6 mA como salida del microcontrolador.

$$R = \frac{V}{i} \quad (2.1)$$

$$R = \frac{3.3-1.5}{6^{-3}} \quad (2.2)$$

$$R = 233 \quad (2.3)$$

Escogiendo una resistencia comercial se utilizará una de 330 Ω

Para que el transistor entre en modo saturación se requiere una corriente de 15 mA, se calcula la resistencia de base considerando el suministro de voltaje para dispositivos periféricos, 3.3 V, voltaje colector emisor del optoacoplador y del transistor

$$R = \frac{V_{cc}-V_{ce \text{ opto}}-V_{base \text{ sat}}}{i} \quad (2.4)$$

$$R = \frac{3.3-0.15-1.2}{15 \text{ mA}} \quad (2.5)$$

$$R = 63 \quad (2.6)$$

Se utiliza la resistencia comercial de 68 Ω

Es indispensable utilizar un diodo en antiparalelo al motor, porque cuando el transistor entra en corte y no conduce, la intensidad comienza a descender y el diodo comienza a conducir produciendo un circuito de extinción de campo magnético motor-diodo. Para este fin se utilizó un diodo 1N5400.

La electroválvula al ser solamente de dos posiciones se controlará con encendido ON/ OFF desde un PIN 12 pero se necesita 78 mA con una alimentación de 3V para cambiar de estado, de acuerdo con la hoja de datos, razón por la cual se utiliza el mismo circuito para la válvula.

2.5.2.2 Control

La conexión del sensor, por las características indicadas en el capítulo 1 y botones se la hará directo a los pines, sin embargo, para conectar los dos esclavos al bus I2C se lo realizara como la Figura 2.5 se presenta el diagrama de conexiones en la Figura 2.6.

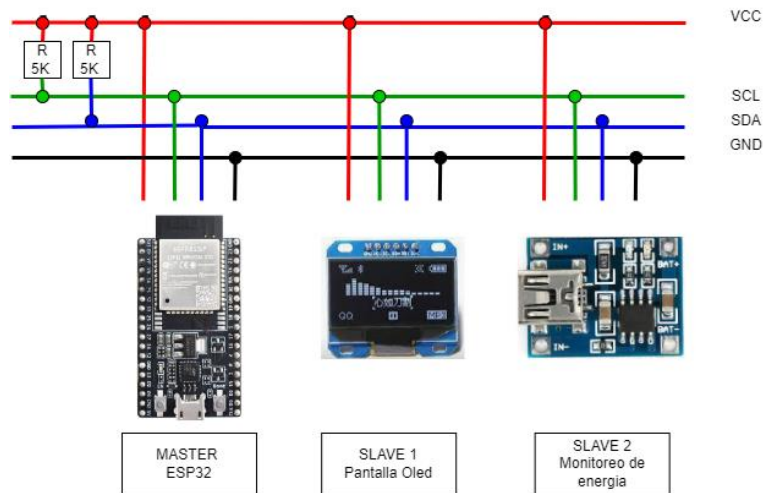


Figura 2.5. Conexión de varios dispositivos a bus I2C

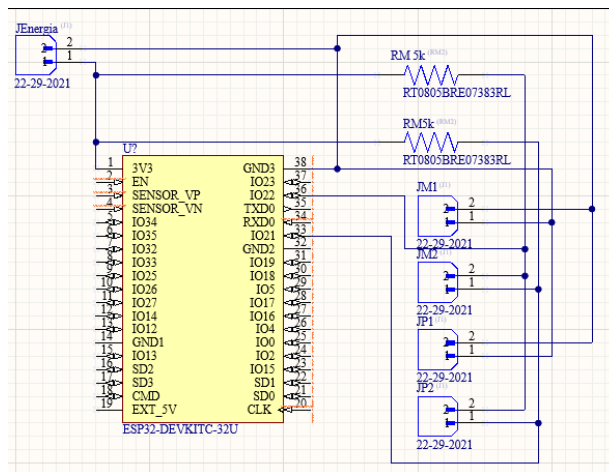


Figura 2.6 Diagrama de conexiones

Se diseñó la placa para que los elementos solo estén 3.5 cm de ancho, con el fin que el circuito de energía y pantalla cubran los otros 3.5 cm del lado izquierdo, para el diseño se debe localizar donde estará el sensor, medida que nos proporciona la tubería neumática y la almohadilla oclusora.

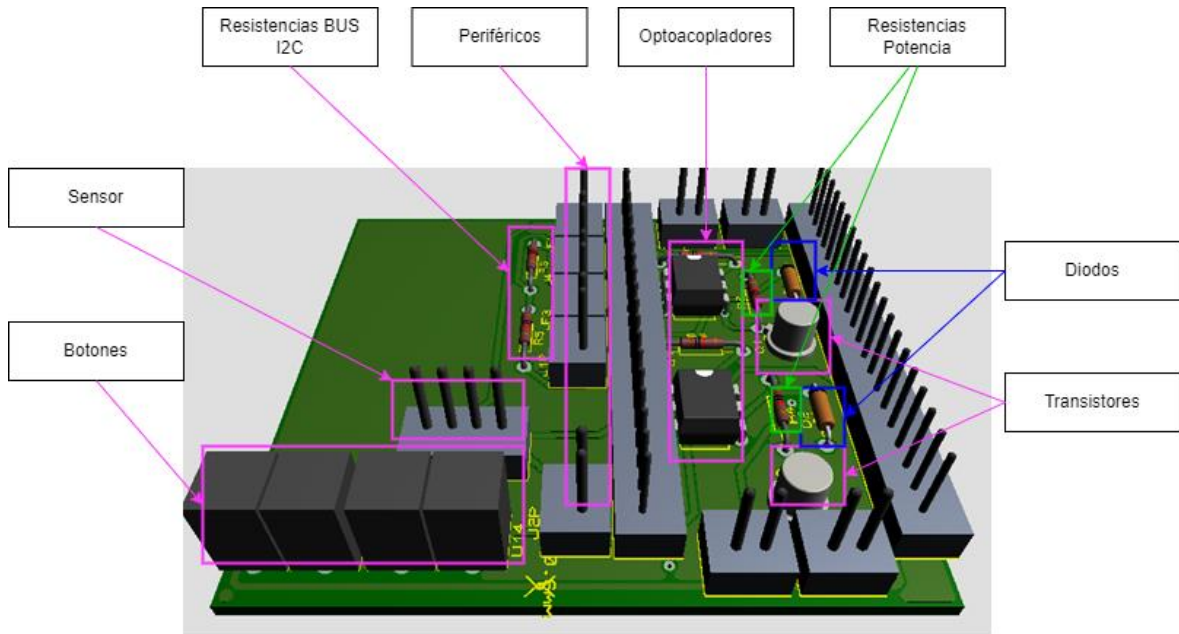


Figura 2.7. Diseño de placa de control

2.6 Diseño del Software:

2.6.1 Software del microcontrolador

La programación del chip ESP32 desarrollado en la plataforma Arduino se compondrá de cuatro subprocesos



Figura 2.8. Diagrama de flujo del software de control del tensiómetro

2.6.1.1 Establecimiento de comunicación

Se proporciona al dispositivo de autonomía para conectarse a una nueva red WIFI, sin la necesidad de subir un nuevo firmware via USB para cambiar y guardar el SSI y password de la nueva red

Lo que se utiliza es una interfaz de transmisión inalámbrica conocida por sus siglas en ingles OTA (Over The Air). En esta, la biblioteca primero establece la conexión WIFI, después crea un servicio web al que se puede acceder si se configura el chip como *Access Point*, luego, se procede a subir una actualización de firmware desde cualquier dispositivo inteligente (celular).

Así, tras tener una conexión exitosa se implementa el protocolo WebSocket, el cual permite una comunicación bidireccional utilizando conexión TCP. Se deja abierta la conexión entre cliente – servidor, de tal forma que se da acceso al usuario para poder subir actualizaciones del firmware. Adicionalmente, el protocolo Web Socket permitiría incorporar servicios IOT para futuras aplicaciones.

Para esta etapa se utilizan las siguientes librerías de Arduino: *WiFiManager*, *AsyncTCP*, *ESPAsyncWebServer*, y *AsyncElegantOTA*.

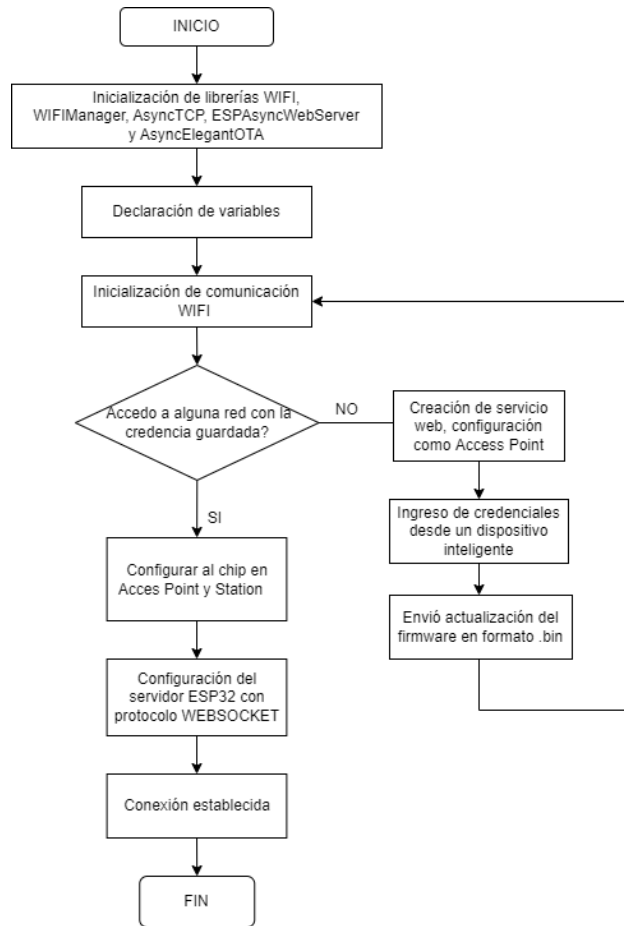


Figura 2.9. Diagrama de flujo para establecer comunicación

2.6.1.2 Adquisición de configuraciones del usuario

En esta segunda etapa se adquieren las configuraciones del usuario para realizar el monitoreo de 1 a 24 horas. Aquí se utilizan 4 botones: *START*, *SUBIR*, *BAJAR* Y *SELECT*, los cuales permitirán el manejo de varios menús.

El botón *START* enciende al dispositivo, e inmediatamente despliega en la HMI el primer menú para elegir al usuario. Tras esto se despliega un segundo menú para escoger entre dos opciones:

1. Tomarse la presión en ese momento y,
2. Configuración e inicio de monitoreo de presión ambulatoria.

Si se escoge la segunda alternativa se podrá elegir de entre las siguientes configuraciones aceptadas: tiempo del estudio (de 1 a 24 horas), tiempo de inicio, tiempo fin del sueño y lapsos de tiempo entre cada monitoreo.

En cada configuración se utiliza los botones de SUBIR y BAJAR para cambiar los tiempos, y tras elegir el valor de tiempo deseado, se puede pasar a la siguiente configuración que se requiera modificar pulsando SELECT.

Por último, en caso de que ya esté programado un MAPA por el usuario, cuando se presione la tecla START se visualizará un menú con dos opciones: la primera “Tomar presión” y la segunda “Re – configurar MAPA”.

Cabe mencionar que una vez que el usuario culmina el ingreso de datos, internamente en el microcontrolador se ejecuta una función para guardar las configuraciones en la memoria EPROM. Esta función también permite eliminar la configuración de MAPA si existiera alguna configuración previa al uso del dispositivo.

La Figura 2.10 se pueden ver los menús configurados, la Figura 2.11 se presenta el diagrama de flujo.

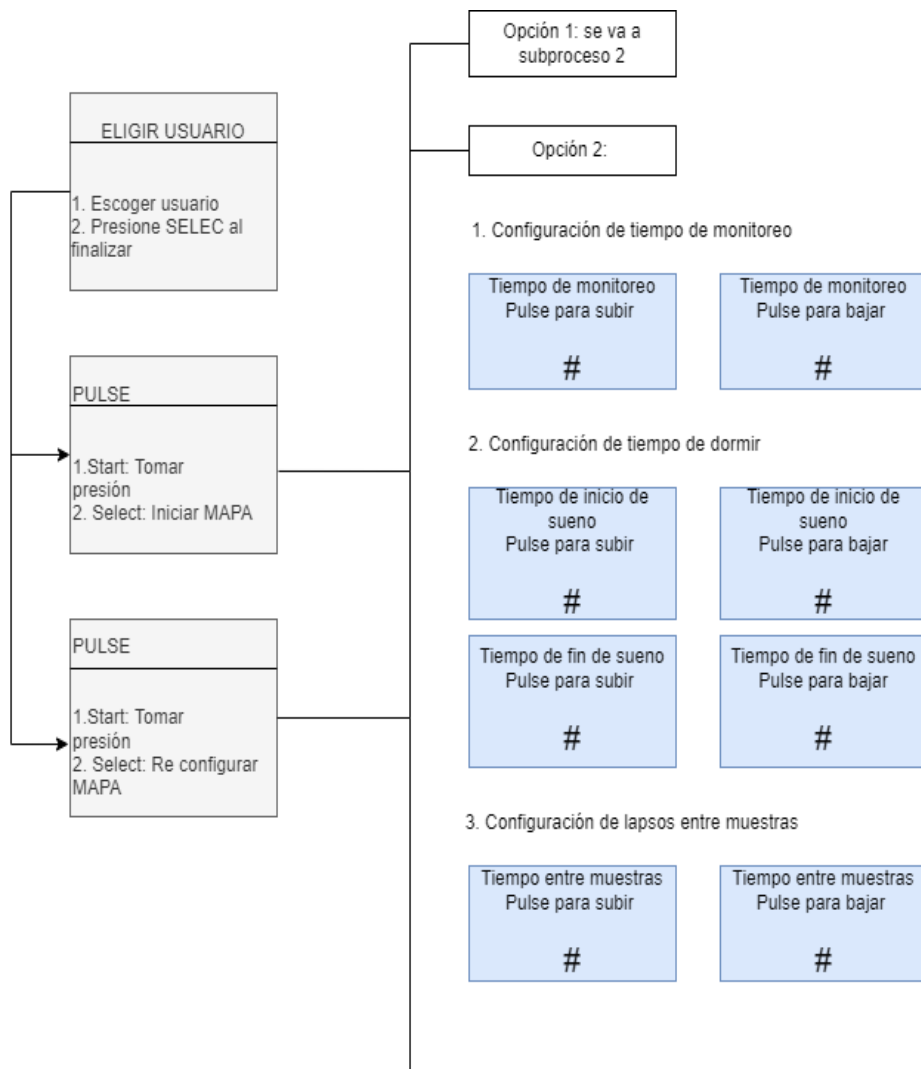


Figura 2.10. Menús para el usuario

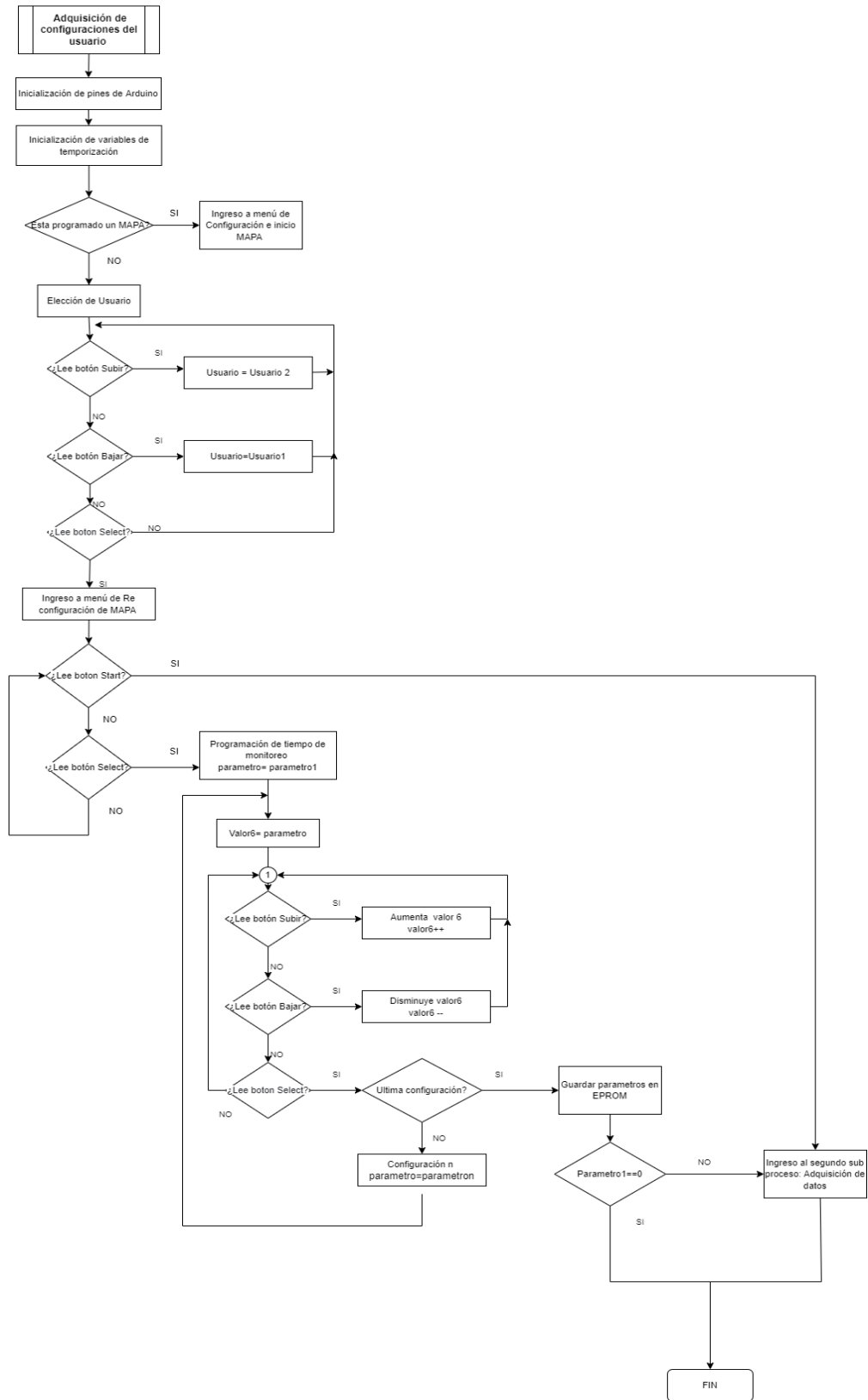


Figura 2.11. Diagrama de flujo de Adquisición de configuraciones del usuario.

2.6.1.3 Adquisición de datos

Se Como tercera etapa, se controla la adquisición de datos basados en el PID del capítulo 1. Aquí se determina el rango máximo de presión en 200 mmHg con base en el valor que manejan las líneas de los dispositivos comerciales mencionados en el primer capítulo. Se controla el ingreso de aire con una señal PWM enviada al motor desde el PIN 27 del microcontrolador para manejar el nivel de corriente que se induce. Se trabaja juntamente con el accionamiento de la válvula, y se envía una señal digital de 1 lógico desde el PIN 12 para cerrar la válvula mientras se alcanzan la presión de los 200 mmHg.

Durante todo este tiempo la presión de la banda de aire es monitoreada para que al llegar al máximo valor programado se envíe una señal PWM que pase de 255 a 0 para que el motor se apague. Por su parte, la válvula se mantendrá cerrada en el rango desde 0 a 200 mmHg de presión y de 200 hasta 50 mmHg. El último valor se tomó considerando también los rangos de los dispositivos comerciales.

Desde el momento en el que inicia este tercer subproceso el sensor de presión se activa, y mide los valores de presión del interior de la banda, sin embargo, los datos útiles para el análisis solo corresponden al intervalo de desinflado o de bajada de presión. Es así como solo en este lapso se configura al dispositivo para que guarde los datos de presión en un archivo tipo JSON.

Para que esta tercera etapa de “Adquisición de datos” inicie se emite una señal directa mediante dos opciones: la primera, al pulsar START, y la segunda, de acuerdo con la programación de monitoreo ambulatorio de presión configurada. Entonces, cuando el usuario está despierto el lapso para la adquisición de datos obedecerá al programado. En el caso de que la hora actual se encuentre dentro del horario de sueño del usuario la adquisición de datos se realizará cada hora y finalizará cuando termine el periodo de estudio.

Entre los intervalos de tiempo no activos de medición de presión se deberá evitar que el microcontrolador permanezca encendido. Es así que una vez que culmina la adquisición de datos el microcontrolador ingresará en estado de sueño profundo. En la Figura 2.12 se visualiza el consumo de energía en cada modo de operación del microcontrolador.

Para este dispositivo se programa que en cada lapso ingrese en modo Deep-sleep con el menor consumo de energía funcionando el tercer procesador de la ESP32, ULP (ultra low power) de bajo consumo de energía que junto a la memoria RTC, temporizadores y sensores internos permite manipular el encendido del núcleo principal.

Power mode	Description	Power consumption
Active (RF working)	Wi-Fi Tx packet 14 dBm ~ 19.5 dBm	Please refer to Table 10 for details.
	Wi-Fi / BT Tx packet 0 dBm	
	Wi-Fi / BT Rx and listening	
Modem-sleep	The CPU is powered on.	Max speed 240 MHz: 30 mA ~ 50 mA Normal speed 80 MHz: 20 mA ~ 25 mA Slow speed 2 MHz: 2 mA ~ 4 mA
Light-sleep	-	0.8 mA
Deep-sleep	The ULP co-processor is powered on.	150 μ A
	ULP sensor-monitored pattern	100 μ A @1% duty
	RTC timer + RTC memory	10 μ A
Hibernation	RTC timer only	5 μ A
Power off	CHIP_PU is set to low level, the chip is powered off	0.1 μ A

Figura 2.12. Consumo de corriente en los diferentes modos del chip ESP32 [21]

Se utiliza dos fuentes que actúan como bandera para despertar al procesador principal del modo de ahorro de energía y proceder a tomar los datos de presión. La primera fuente es manipulando el tiempo de un temporizador RTC, también denominado definidor del tiempo del sueño, el cual permite que al finalizar un contador programado salga del modo Deep Sleep y el microcontrolador de nuevo encienda todas sus funciones, el contador o tiempo para dormir estará en función directa de las configuraciones que el usuario establezca; la segunda forma es programando una interrupción externa para salir del modo Deep Sleep la misma que se configura en el botón START dándole facultades de prender la ESP y proceder a tomar presión sin importar la configuración existente de monitoreo ambulatorio de presión.

El temporizador RTC tiene varias fuentes de reloj internas y la posibilidad de conectar un cristal externo dándole la capacidad de mantener un contador de hora durante los modos de suspensión. En la Figura 2.13 se pueden apreciar que todas las funciones importantes del chip están apagadas en el modo Deep Sleep, y por tanto todas las variables declaradas en configuraciones se hacen cero cuando se ingresa a este modo. Por tal razón en la etapa 2 se crea una función para guardar las configuraciones de usuario en la memoria EPROM.

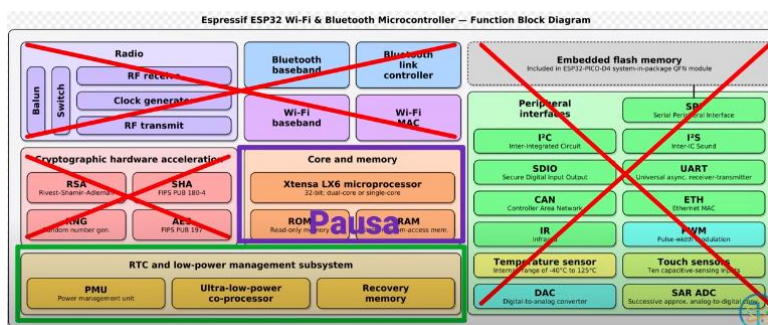


Figura 2.13. Funciones activas durante el modo Deep Sleep.

En el presente diagrama de flujo se observa cómo se ejecuta el código.

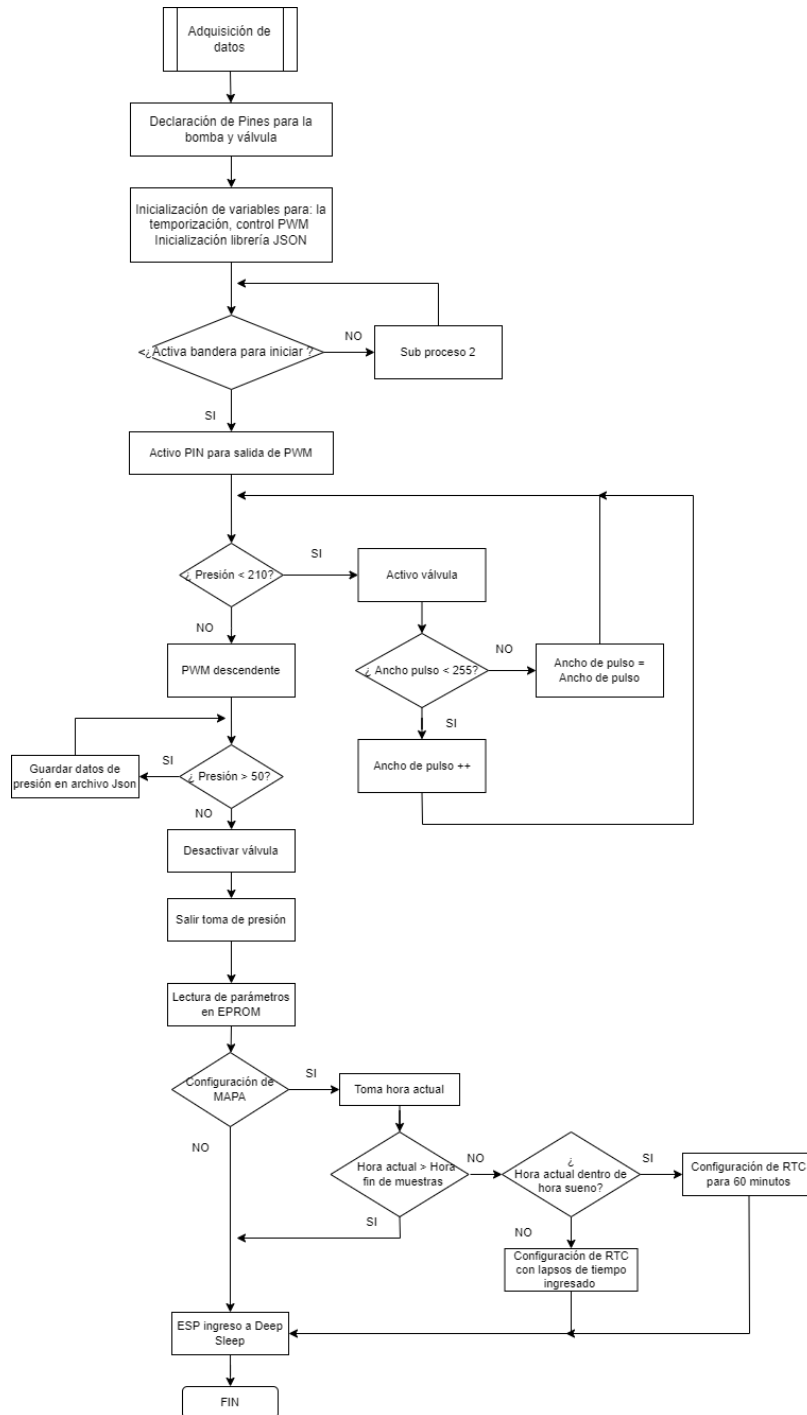


Figura 2.14. Diagrama de flujo de Adquisición de datos

Se presenta un segundo diagrama de flujo con el fin aclara la programación para el establecimiento de lapso entre cada medición, cabe recalcar que entre los lapsos de tiempo se envía dormir al dispositivo, cuando finaliza la “Adquisición de datos” se levanta una bandera la cual es utilizada para enviar datos y enviar a dormir al dispositivo. Así, se lee las configuraciones guardadas en la memoria EPROM de la etapa 1, en la función establecida para guardar configuraciones del usuario. También se guarda la hora en la que

se configuro el MAPA, y se establece un tiempo cero desde que inicia el MAPA. Es importante mencionar que se convierte a minutos a todas las configuraciones necesarias, por ejemplo:

INICIO DE SUENO:

Dato guardado en memoria Eprom: 23

Conversión: $23 \times 60 = 1380$

De esta forma, se determina en minutos cuanto tiempo debe transcurrir desde el inicio del MAPA hasta llegar a inicio y fin de sueño. Después de cada toma de muestras se adquiere la hora actual y con dicha hora se analiza cuanto tiempo ha transcurrido desde que inicio el MAPA hasta la hora actual.

Si el tiempo transcurrido es menor al tiempo de estudio entra a condiciones para determinar si el usuario está despierto o dormido. En caso de que el tiempo este dentro la hora de sueño se envía a dormir el chip por 60 minutos, caso contrario si el usuario está despierto el tiempo configurado para que el chip duerma vendrá directamente de las configuraciones del usuario "Lapso de tiempo entre muestras".

Cabe señalar que como se otorgó al botón START el permiso para tomar la presión en cualquier momento que necesite el usuario la calibración de la hora del sueño se puede desequilibrar. Esto ocasionaría que el equipo se prenda, en el peor escenario, una hora más tarde a pesar de que ya haya terminado el lapso del sueño y el tiempo de estudio.

Considerando esto se incluyó una última condición: se compara la hora de sueño y el tiempo que falta para llegar al fin del sueño. Si el resultado es mayor a 60 minutos el tiempo de sueño será 60 minutos. Por otro lado, si el resultado es menor a 60, pero mayor a 15 minutos, el tiempo de sueño se define como el valor que reste para llegar a la hora de despertarse. Por último, si el tiempo restante para despertarse es menor a 15 minutos se programa el tiempo de activación en 15 minutos.

Se utilizó esta misma lógica para establecer el último tiempo en que el usuario esté despierto ya que si el usuario configura tiempos mayores a 25 minutos se puede perjudicar el número de muestras necesarias de tomar en la noche.

Por último, también se compara si el tiempo transcurrido desde el inicio del MAPA es mayor al "Tiempo de estudio". En ese caso se encera a todas las configuraciones guardadas en

la memoria EPROM y se envía a dormir indefinidamente al dispositivo esperando una fuente de activación.

Las líneas de código a continuación permiten configurar el tiempo durante el cual el chip entra en sueño profundo o Deep Sleep

```
esp_sleep_enable_timer_wakeup(sleepTime1)
```

```
esp_deep_sleep_start()
```

En esta etapa se emplean algunas variables almacenadas en la memoria RTC para determinar tiempos. La Figura 2.15 presenta el diagrama de flujo desglosado de este subproceso.

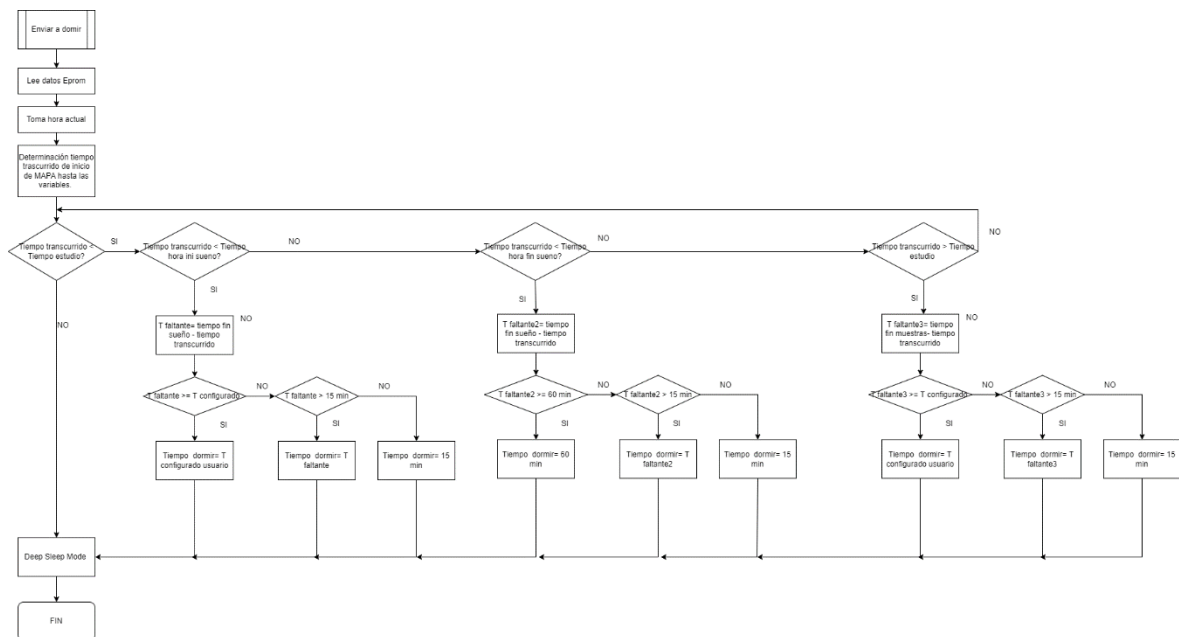


Figura 2.15. Diagrama de flujo para configurar el tiempo de sueño.

2.6.1.4 Envió, recepción y almacenamiento de datos

El objetivo del subproceso es el envío de datos de presión en tiempo real para su posterior análisis y almacenamiento. En el subproceso 3 se creó el archivo en formato JSON que contiene los datos necesarios para el análisis, dado que se utilizará una API REST para conexión entre cliente servidor se utilizará el protocolo HTTP para el envío de datos. Inmediatamente se termine la recolección de datos se levanta una bandera que abre el protocolo serializa al documento JSON y envía los datos por medio de una solicitud POST, el protocolo HTTP no está orientado a sesión, la conexión se cierra inmediatamente tenga respuesta del servidor, por ende, una vez enviado el documento se debe cerrar la sesión.

Es importante incluir los datos de presión en un archivo JSON permite enviar un gran número de información en un solo POST y evita la pérdida de datos por la velocidad del WIFI, pues los datos arrojados por el sensor están en el orden de los milisegundos.

Las librerías utilizadas son HTTPClient y JSON.

Los datos se almacenan en Mysql.

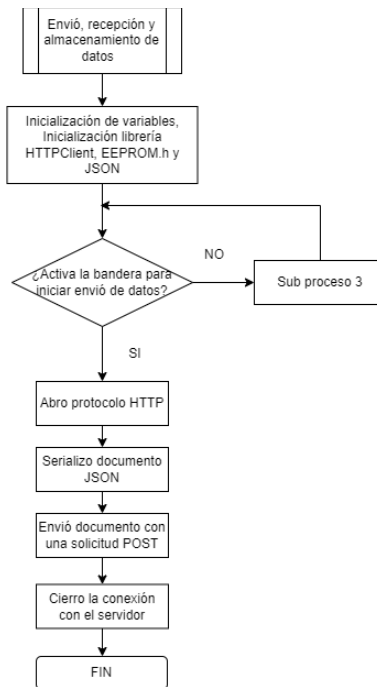


Figura 2.16. Diagrama de flujo envío de datos

2.6.2 Software utilizado para el envío de datos en tiempo real.

2.6.2.1 API REST

Como ya se mencionó la API REST permite la recepción y administración de datos en tiempo real y es la tecnología preferida si se requiere menores tiempos de latencia.

La Figura 2.17 se puede ver la interacción entre ESP32, API REST, MySQL y Matlab, los datos de envío y recepción desde la ESP32 son en formato tipo JSON.

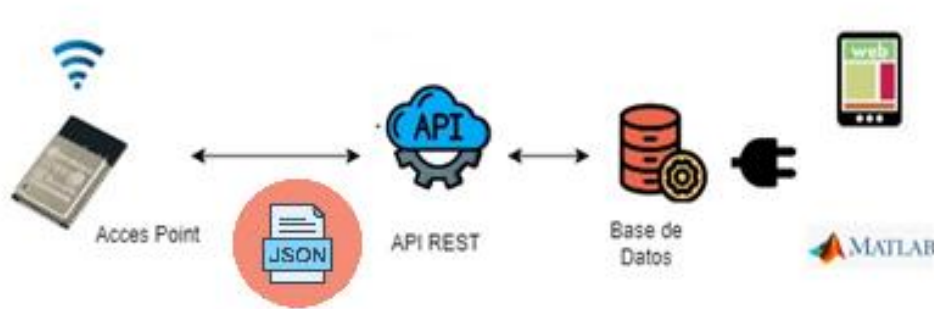


Figura 2.17. Interacción entre ESP32 y API REST

Se programo en Lenguaje PYTHON una API REST utilizado el firmware Flask permite administrar los recursos de llegada, agrega nuevos end points y utilizar decoradores o rutas por medio de `@app.route`, para realizar solicitudes y respuesta HTTP para la comunicación cliente servidor se utiliza la herramienta `requests` y `response`, para realiza solicitudes HTTP se utiliza los verbos POST y GET.

```
from flask import Flask, jsonify, request
@app.route('/datos', methods=['GET'])
def listar_datos():
    try:
        cursor = conexion.connection.cursor()
    @app.route('/datos', methods=['POST'])
    def registrar_dato():
```

Para conectar a la API con la base de datos debemos importar recursos con MySQL con los códigos

```
from flask_mysql import MySQL
```

Se utiliza el conector SQL para llamar a la base de datos y tabla específica.

```
sql = "SELECT dato1 FROM datos ORDER BY numero DESC LIMIT 380"
```

Es importante agregar el paquete `jsonify` ya que los datos tanto de llegada como de salida a la ESP32 estan en formato tipo JSON.

```
from flask import Flask, jsonify, request
```

2.6.3 Software utilizado para el preprocesamiento de la información.

Para realizar un preprocesamiento de los datos se utiliza Matlab una vez levantada la API REST se intercambia información con solicitudes HTTP demostrando que cualquier dispositivo cliente que tenga integrado el protocolo puede hacer uso de la API.

Se envía una solicitud GET para acceder a los datos, Matlab procesa los datos y se regresa el resultado con una solicitud POST. Se utiliza el formato tipo JSON para envío y recepción, como ya se mencionó su formato es fácil de comprender y cualquier lenguaje puede utilizarlo.

En la Figura 2.18 se mira la dinámica entre API REST, Matlab y base de datos.

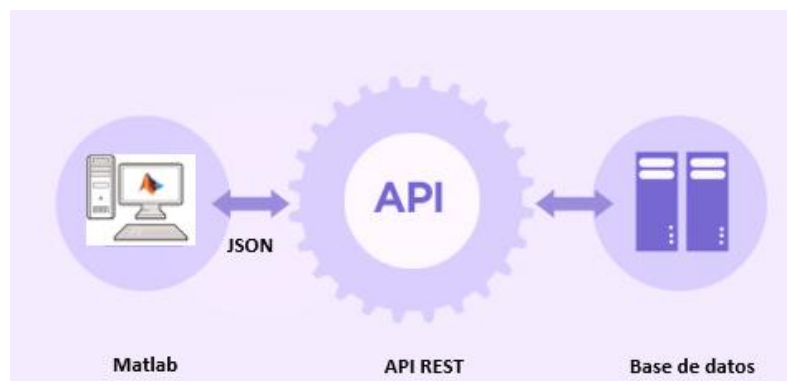


Figura 2.18. Esquema de conexión entre Matlab, API REST y base de datos

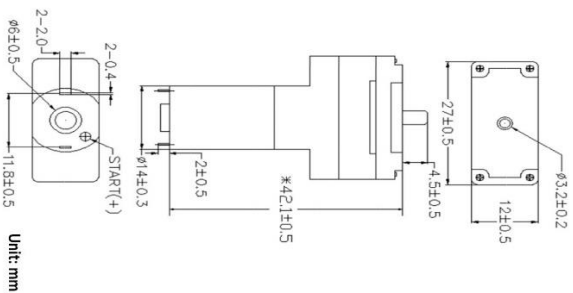
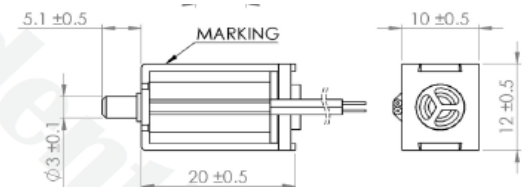
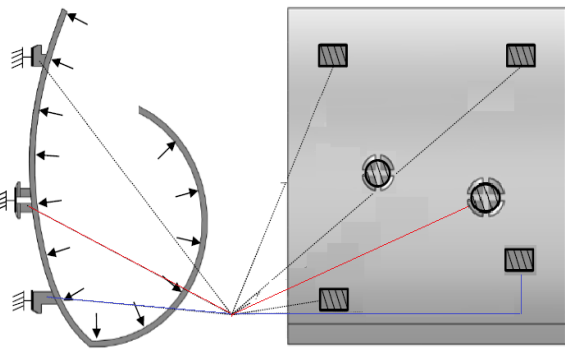

2.7 Diseño de carcasa 3D

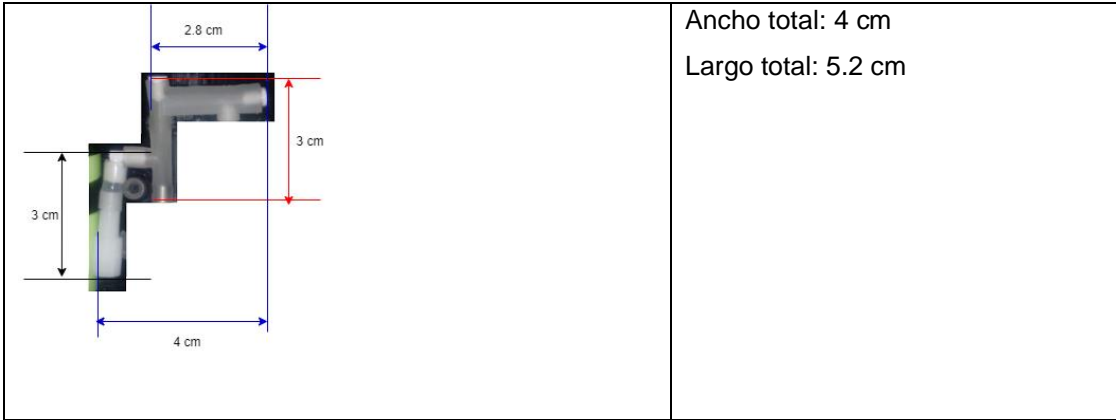
El diseño de la carcasa se lo realiza con el software FreeCAD ya que no requiere un diseño complejo, su interfaz es sencilla y es gratuito.

Primero, se dimensiona la carcasa del tensiómetro con las medidas de los elementos que contiene la base del tensiómetro. En la Tabla 2.4 se hace un listado y de las dimensiones de los elementos que contendrá la carcasa.

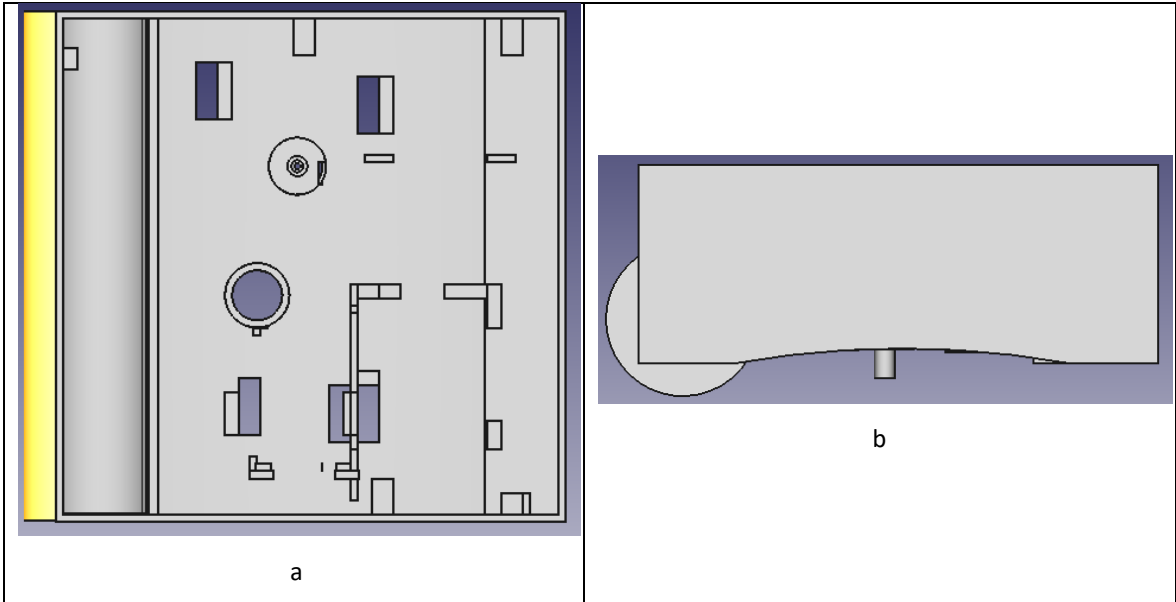
Tabla 2.4 Dimensiones de elementos

Elemento	Dimensiones
----------	-------------

<p>Dimensiones de la Bomba KMP14A</p>  <p>Unit: mm</p>	<p>Largo: 4.2 cm Largo del motor: 1.6 cm Ancho de la bomba: 1.4 cm Largo de la bomba: 2.6 cm Alto: 1.2 cm</p>
<p>Dimensiones de la electroválvula KSV05</p> 	<p>Largo: 2 cm Alto: 1 cm Ancho: 1.2 cm</p>
<p>Dimensiones de la banda de aire</p> 	<p>Espacio entre las agarraderas: Ancho entre las dos primeras: 2.1 cm Ancho entre las secundarias: 2.6 cm Largo: 4.3 cm entre las primarias y secundarias Posiciones en XYZ para el tubo de aire del sensor X: 1.8 cm; Y: 2.4 cm ; Z: 4 mm Posición en XYZ para el tubo de ingreso del aire. X: 2.4 cm; Y: 4.1 cm ; Z: 4 mm</p>
<p>Batería recargable</p> 	<p>Diámetro: 1.8 cm Altura: 6.5 cm</p>
<p>Dimensiones de conducto Neumático</p>	<p>Dimensiones</p>



Para el diseño se tomara en cuenta todas las dimensiones de la Tabla 2.4 y la curvatura que presenta la banda de aire es de 3mm, medida tomada desde un plano horizontal de la banda hasta el punto máximo de la curvatura, se debe tomar en cuenta para el diseño caso contrario las agarraderas no ingresaran en los huecos. Es fundamental que el diseño de la carcasa tenga exactamente las mismas medidas indicadas de la banda de aire proporcionando la posición exacta del sensor y de la salida de aire, las dimensiones de la banda de aire también influyen el tamaño y diseño de la placa de control.



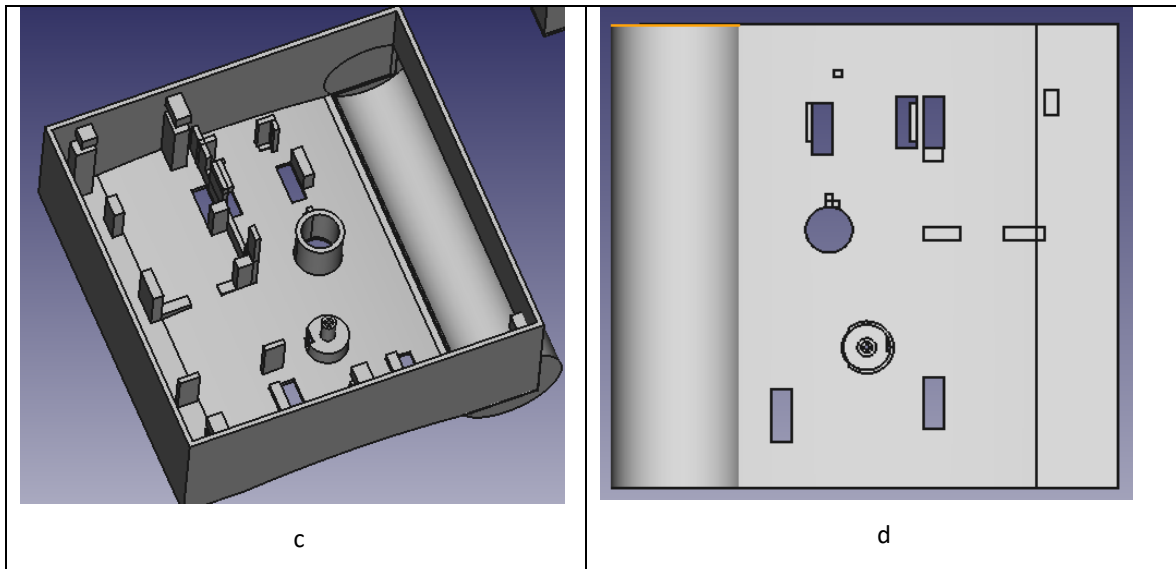


Figura 2.19. a) Vista de base b) Vista lateral c) Vista superior d) Vista de planta de la carcasa

Acorde a las dimensiones de la bomba, válvula y pila se agregó topes para cada uno, cobrando mayor importancia topes de la válvula y bomba esto evita que se muevan y existan fugas de aire

Para la tapa primero se dimensiono la placa electrónica y los elementos periféricos como botones y pantalla OLED

Dimensiones de la placa electrónica: 6.5 x 6.5 x 2.5 cm.

Dimensiones pantalla OLED: 3,50 x3,23x1,14 cm.

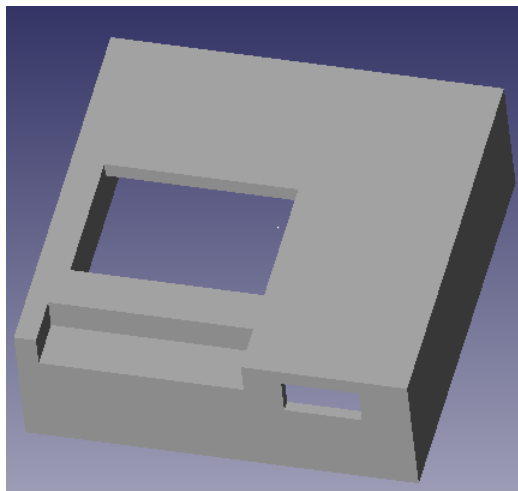


Figura 2.20. Tapa de la carcasa

3. Resultados, conclusiones y recomendaciones

1.8 Resultados

En esta sección primero se presenta los resultados de la calibración del sensor y se procede a exponer el Hardware del prototipo empezando por la placa electrónica y carcasa diseñada, por último se presenta los resultados del Software para el control, por medio de fotos se visualizara los menús programados para receptar configuraciones de usuario, se comprueba la automatización de la toma de presión con la recolección de datos de presión en la API REST, se valida el funcionamiento de la API por dos punto el primero la ejecución de POST y GET utilizando MATLAB y el segundo verificando que los datos fueron guardados en MySQL.

3.1.1 Resultados de calibración.

Se presenta las gráficas como el resultado de la calibración, en la primera grafica se observa la salida digital del sensor aplicando variaciones de 10 mmHg, desde 0 hasta 210 mmHg observando se determina se puede hacer un ajuste lineal, se utilizó el método de los mínimos cuadrados para ajusta la curva, en la Figura 3.2 se observan los resultados, las mayores desviaciones del patrón comienzan a partir de 140 mmHg, la curva amarilla identificada como la salida del sensor tiene variaciones de +- 6 mmHg, la curva naranja y ploma representan al resultado de la calibración tanto de subida como de bajada tienen variaciones de +- 1 y +-2 mmHg respectivamente comparados con la curva del tensiómetro manual y digital.

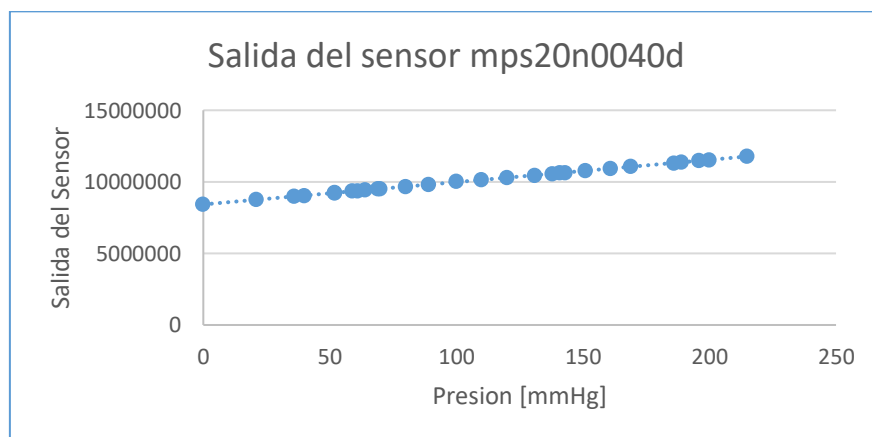


Figura 3.1. Respuesta del sensor para cambios de presión.

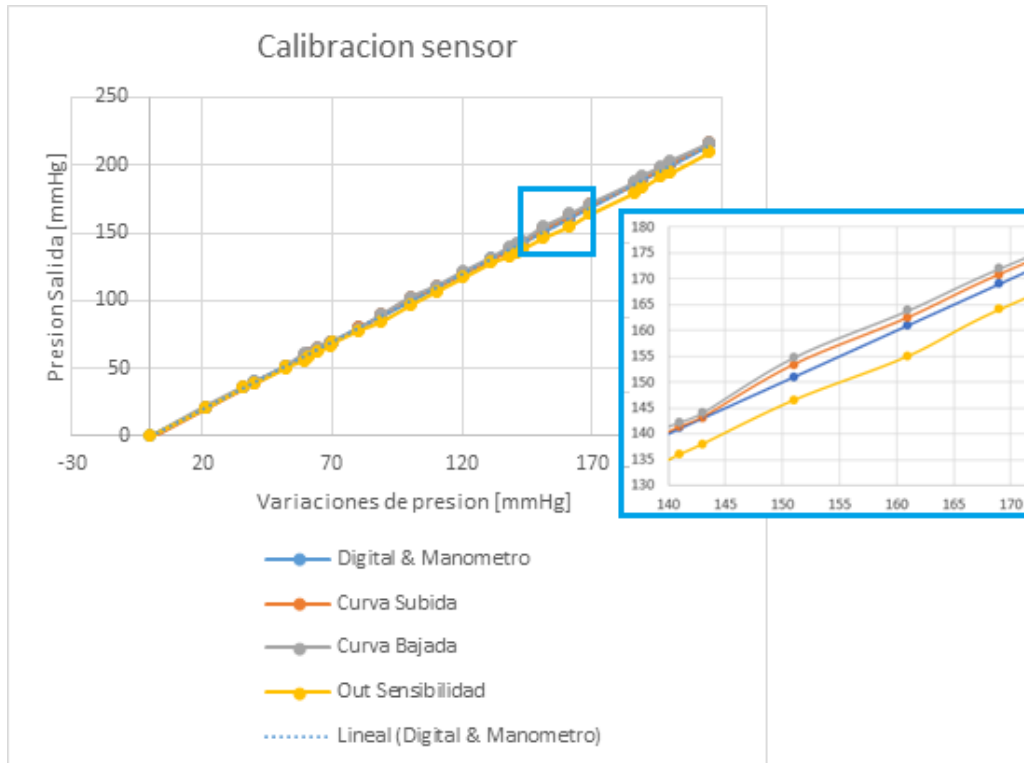


Figura 3.2. Curvas de presión de manómetro, tensiómetro digital y sensor

La ecuación 3,1 representa la linealización de la curva de bajada, comparando con la salida del tensiómetro digital tiene variaciones máximas a +- 1 mmHg.

$$\text{Presion} = \text{OUTSensor} * 6.48054437 * 10^{-546.4210747} - 546.4210747 \text{ [mmHg]} \quad (3.1)$$

3.1.2 Resultados de Hardware.

3.1.2.1 Placa electrónica

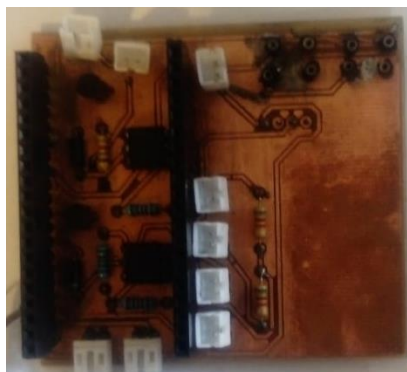


Figura 3.3. Placa de potencia y control

3.1.2.2 Carcasa



Figura 3.4. Carcasa del prototipo

Se comprueba que el dimensionamiento de elementos dentro de la base de la carcasa fue eficiente, la Figura 3.5 se puede comprobar que los orificios para el sensor y tubo neumático están de acuerdo con las medidas del brazalete, los topes creados inmovilizan a la válvula y bomba, existe la longitud exacta para la inclusión de la batería de litio evitando interferir con el sistema neumático, la curvatura permite que el brazalete se adhiera a la carcasa.



Figura 3.5. Carcasa a) Vista de la base con los elementos incluidos b) Vista lateral, brazalete adherido

3.1.3 Resultados de Software en el dispositivo.

3.1.3.1 Establecer conexión WIFI

El dispositivo cuenta con autonomía para acceder a una nueva red WLAN sin tener que cambiar el código y subirlo por cables, el usuario puede configurar el nombre de la red y contraseña desde su teléfono celular. Se procede a verificar el funcionamiento.

Cuando se enciende el dispositivo busca las redes disponibles, si no tiene una red de de WIFI y contraseña guardados inmediatamente se configura como punto de acceso. La ESP estara esperando que accedan a su RED, denominada AutoConnectAp, como se puede ver en la Figura 3.6 al no haber dispositivos conectados la respuesta de NUM CLIENTS es 0.

```
COM3
18:21:01.005 -> *wm: [2] [EVENT] WIFI_REASON: 201
18:21:01.005 -> *wm: [2] [EVENT] WIFI_REASON: NO_AP_FOUND
18:21:01.005 -> *wm: [2] Connection result: WL_NO_SSID_AVAIL
18:21:01.052 -> *wm: [1] AutoConnect: FAILED
18:21:01.052 -> *wm: [2] Starting Config Portal
18:21:01.052 -> *wm: [2] Disabling STA
18:21:01.052 -> *wm: [2] Enabling AP
18:21:01.052 -> *wm: [1] StartAP with SSID: AutoConnectAP
18:21:01.052 -> *wm: [2] AP has anonymous access!
18:21:01.572 -> *wm: [1] AP IP address: 192.168.4.1
18:21:01.572 -> *wm: [1] Starting Web Portal
18:21:01.572 -> *wm: [2] HTTP server started
18:21:01.572 -> *wm: [2] Config Portal Running, blocking, waiting for clients...
18:21:20.230 -> *wm: [2] NUM CLIENTS: 0
```

Figura 3.6. Configuración como Access Point.

El Usuario visualiza la red AutoConnectAp desde su teléfono celular se conecta y configura la nueva red y contraseña



Figura 3.7. Red AutoConnectAp abierta.

La Figura 3.8 se puede ver las redes WIFI disponibles desde el teléfono celular.



Figura 3.8. Redes WIFI disponibles.

Se elige la red y se escribe la contraseña.

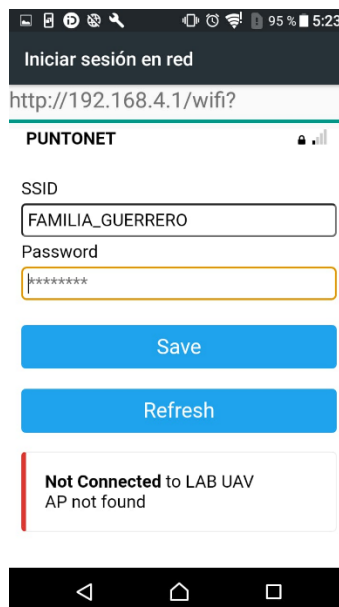


Figura 3.9. Configuración de una nueva red WIFI desde el celular.

En el serial se puede observar que el chip tiene acceso a una red WIFI inmediatamente se configura como Station

```

18:27:02.639 -> *wm:[2] setSTAConfig static ip not set, skipping
18:27:02.639 -> *wm:[1] Connecting to NEW AP: FAMILIA_GUERRERO
18:27:02.686 -> *wm:[1] connectTimeout not set, ESP waitForConnectResult...
18:27:09.680 -> *wm:[2] Connection result: WL_CONNECTED
18:27:09.680 -> *wm:[1] Connect to new AP [SUCCESS]
18:27:09.680 -> *wm:[1] Got IP Address:
18:27:09.680 -> *wm:[1] 192.168.1.17
18:27:09.680 -> *wm:[2] shutdownConfigPortal
18:27:09.680 -> *wm:[2] restoring usermode STA
18:27:10.713 -> *wm:[2] wifi status: WL_CONNECTED
18:27:10.713 -> *wm:[2] wifi mode: STA
18:27:10.713 -> *wm:[2] configportal closed
18:27:10.713 -> *wm:[1] config portal exiting
18:27:10.713 -> connected...yeey :)

```

Figura 3.10. Dispositivo conectado a una nueva Red WIFI

Utilizando el protocolo WEB SOCKET, posibilita subir una actualización del firmware sin cables de conexión, ingresando a la red de ESP /update se comprueba que está activo el protocolo y se puede subir una actualización del firmware On Air.



Figura 3.11. Activo protocolo Web Socket.

3.1.3.2 Configuraciones de usuario

1. Siempre que se ingrese al dispositivo si ya se tiene una red WIFI configurada se visualizara la pantalla de inicio

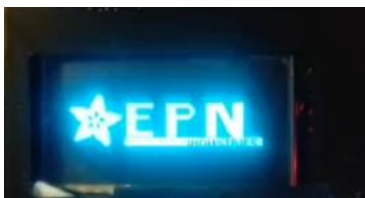


Figura 3.12. Pantalla de inicio

2. Si solo se energiza al dispositivo se visualiza la pantalla de la Figura 3.13 indicando que ya se accedió a una red WIFI y puede empezar a funcionar el tensiómetro

inmediatamente el dispositivo entra en modo Deep Sleep, caso contrario si se presiona la tecla START seguirá a la siguiente pantalla.

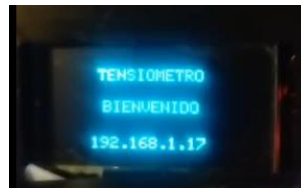


Figura 3.13. Pantalla de inicio

3. Se despliega el menú de elección de usuario se le indica al paciente que primero elija el usuario presionando la tecla de USER 1 o USER2 cuando termine presione la tecla SELECT para guardar la configuración. La Figura 3.14 se puede ver la visualización de las pantallas

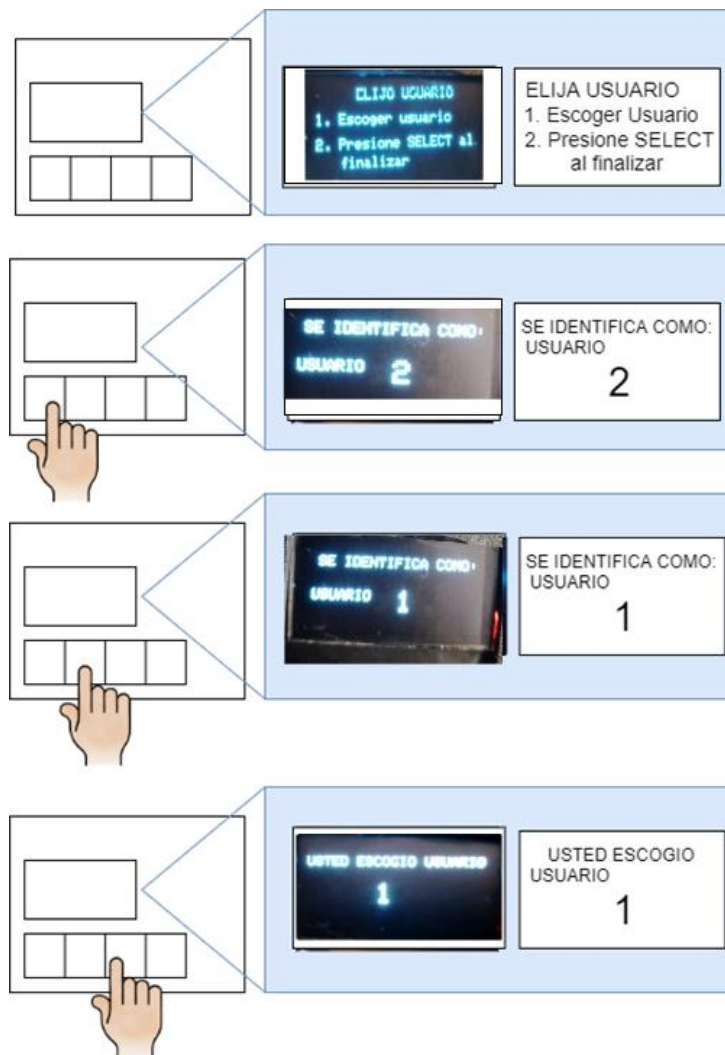


Figura 3.14. Selección de usuario, visualización de pantallas

4. Inmediatamente se ingresa al siguiente menú que indica con la tecla START puedo tomar la presión si se escoge esta opción se toma la presión y va a modo Deep Sleep, si selecciona SELECT procedo a configurar una MAPA, configuro Tiempo de estudio con las teclas USER1 + y USER 2- voy subiendo y bajando el tiempo, para seleccionar y pasar al siguiente menú presiono SELECT

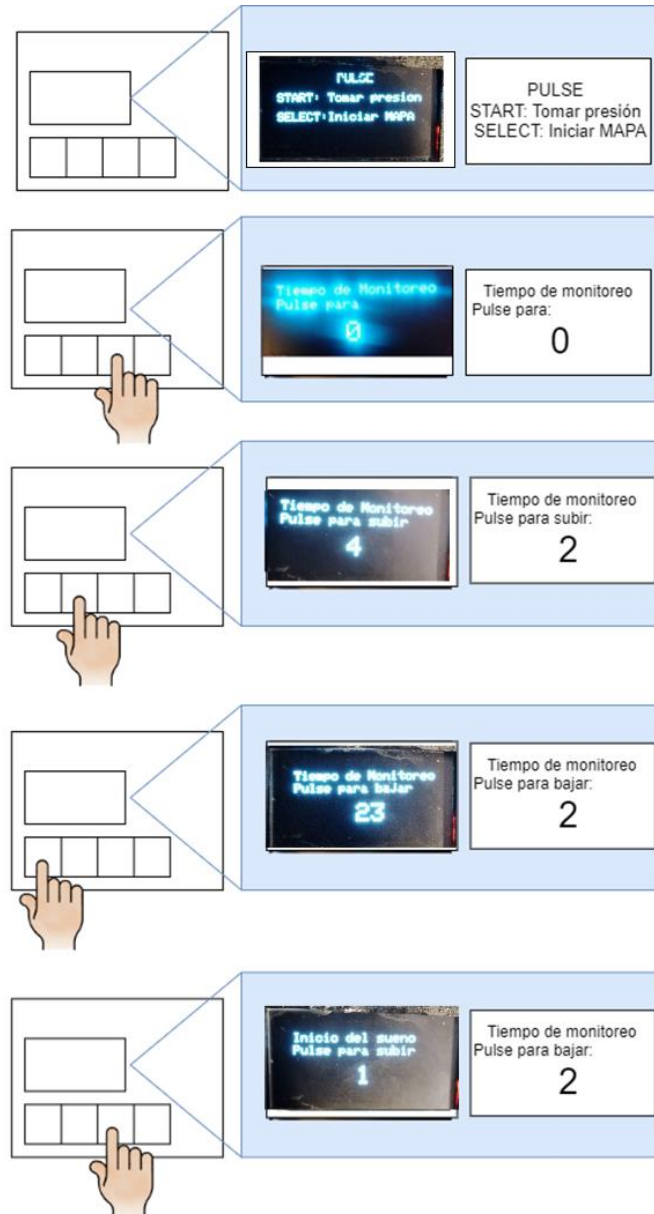


Figura 3.15. Configuraciones del usuario para MAPA

Se utiliza el mismo procedimiento para la configuración de los siguientes parámetros como Hora de inicio, Hora de fin de sueño y lapsos de tiempo entre cada muestra, al finalizar se guardan las variables en la memoria EPROM como se visualiza en la Figura

3.16, el prototipo toma la presión establece que tiempo debe dormir como se explicó en el capítulo dos he ingresa en modo Deep Sleep.

```

06:34:17.020 -> 1
06:34:17.020 -> 987
06:34:17.020 -> 1407
06:34:17.020 -> 900000000
06:34:17.020 -> Going to sleep now
06:34:17.020 -> 10
06:34:17.020 -> 333
06:34:17.020 -> 11111111111111111111
06:34:17.020 -> 1320
06:34:17.020 -> 222222222222
06:34:17.020 -> 300
06:34:17.020 -> 333333333333
06:34:17.020 -> 15
06:34:17.020 -> 7777777777777777
06:34:17.020 -> 1440

```

Figura 3.16. Configuraciones guardadas en memoria EPROM

5. Si se presiona la tecla START y existe un mapa previamente configurado se visualiza la siguiente pantalla

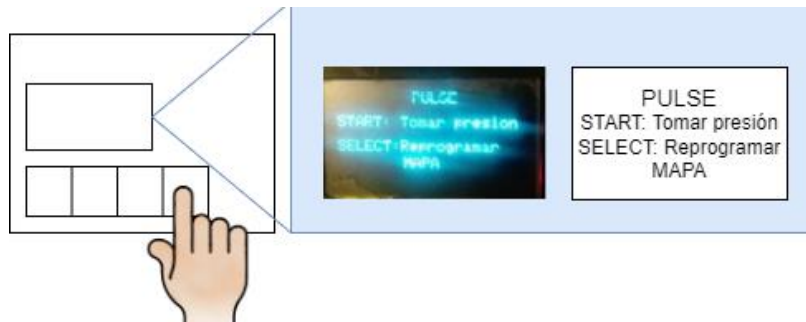


Figura 3.17. Menú si existe previamente configurado un MAPA

Pulsando START tomara la presión y se reanudara el MAPA.

En caso de que presione la tecla SELECT para reconfigurar MAPA el usuario puede eliminarlo colocando la primera variable en 0, se borran los parámetros de la memoria Eprom y el dispositivo entrara en Deep Sleep indefinidamente. En la Figura 3.18 se observa como se puede eliminar un MAPA.

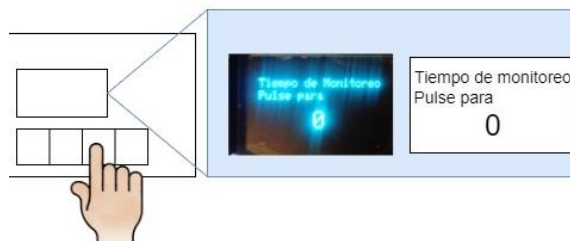


Figura 3.18. Eliminación de MAPA

3.1.3.3 Resultados de Software para él envió, almacenamiento y preprocesamiento de datos.

En Figura 3.19 se puede observar el arribo de datos en un solo paquete JSON y el código 200 que indica la respuesta de conexión y envío de datos fue satisfactoria, por tanto permite verifica el funcionamiento optimo del WIFI configurado de la ESP32, el envío de datos con el protocolo HTTP y el funcionamiento de la API REST que trabaja en conjunto con la base de datos.

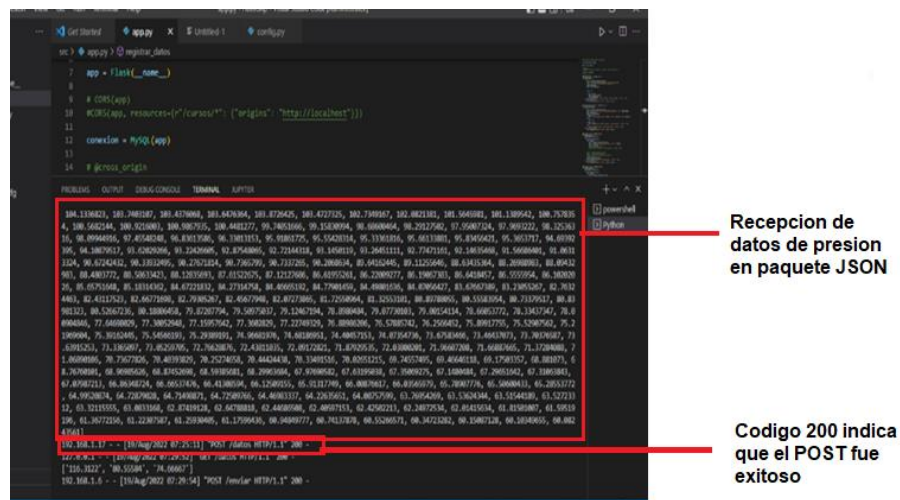


Figura 3.19. Consola de API REST

Se utiliza la herramienta Matlab para el pre procesamiento, se crea una aplicación con dos ventanas, utilizando GET recojo los datos de presión desde la API, se procesa y se envía el resultado con un POST y los datos son almacenados en la base de datos



Figura 3.20. Caratula de la aplicación desarrollada en Matlab

El programa se ha codificado para trabajar en seis etapas, primero recibe los datos desde la base de datos en MySQL a través de la lectura del fichero generado por esta última, con los datos recogidos se procede a la segunda etapa en la que se filtran los datos aislando los valores desde que la manga alcanza la presión máxima configurada hasta que su presión decae por completo, con los datos obtenidos inicia la tercera etapa en la que mediante el cálculo de una curva de tendencia central se identifican las oscilaciones producidas por la presión en los vasos sanguíneos, la cuarta etapa se utiliza la técnica de la máxima amplitud mencionada en el capítulo 1, se estima presión sistólica y diastólica utilizando las ecuaciones 1.2 y 1.3, una vez obtenidas ambas presiones se procede a calcular los valores de ritmo cardiaco con un análisis de picos, en la quinta etapa el programa muestra los datos procesados en la interfaz, indicando todas las variables de interés y la curva de presión real y presión oscilatoria.

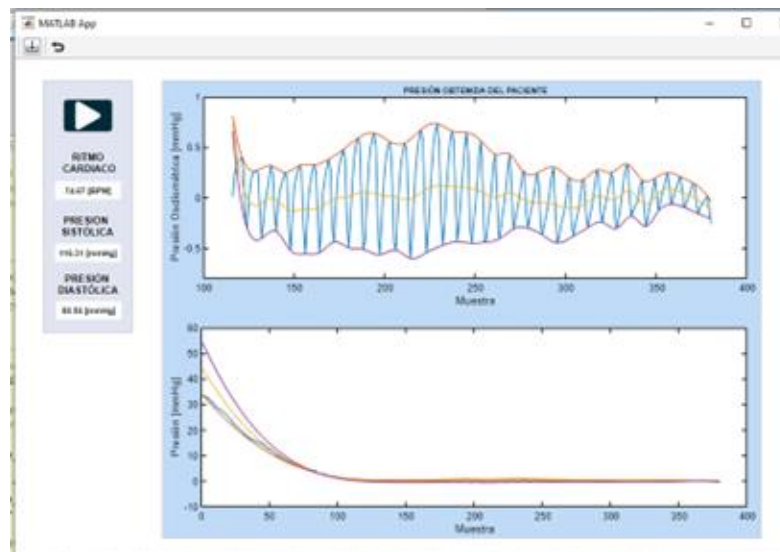


Figura 3.21. Oscilograma procesado en Matlab

La Figura 3.22 corrobora que los datos fueron extraídos de la API con un GET y que se regresó los resultados con un POST

```
196, 61.367/2156, 61.2238/587, 61.2593/405, 61.1759/436, 60.9484/777, 60.7413/878, 60.5526/143561]
192.168.1.17 - - [19/Aug/2022 07:25:11] "POST /datos HTTP/1.1" 200 -
127.0.0.1 - - [19/Aug/2022 07:29:52] "GET /datos HTTP/1.1" 200 -
```

Figura 3.22. Respuesta del protocolo HTTP.

La Figura 3.23 se puede visualizar que los datos fueron almacenados en la base de datos.

	dato1	numero	fecha9	presionalta	presionbaja
	60.33200011	17611	2022-08-19 07:29:11	0	0
	60.34723282	17612	2022-08-19 07:25:11	0	0
	60.15087128	17613	2022-08-19 07:25:11	0	0
	60.10349655	17614	2022-08-19 07:25:11	0	0
	60.08243561	17615	2022-08-19 07:25:11	0	0
		17616	2022-08-19 07:29:54	116	0
		17617	2022-08-19 07:29:54	81	0
		17618	2022-08-19 07:29:54	75	0

Figura 3.23. Datos guardados en MySQL

3.2 Conclusiones

Acorde con los resultados mostrados y con varias pruebas de funcionamiento del control y adquisición de parámetros del usuario se concluye.

- Se desarrollo el diseño de la carcasa en 3D con las dimensiones de todos los elementos contenidos en el prototipo.
- Se realizo la implementación del sistema de control para el prototipo del tensiómetro cumpliendo con los requerimientos de energía para los elementos periféricos que maneja.
- Se desarrollo algoritmos de control para el chip ESP32, los cuales, logran automatizar el proceso de medición de presión, cumpliendo con los requerimientos de tener un control de 24 horas, controlar los lapsos de tiempo entre cada muestra verificando si el usuario esta despierto o dormido.
- Se implemento la Interfaz intuitiva en el dispositivo, que permite adquirir las configuraciones del usuario acompañado de varios menús para visualización de la adquisición de parámetros.
- Se realizó la programación necesaria para que el prototipo admita una tomar valores de presión incluso si existe configurado un monitoreo de presión, evitando interferir con los parámetros programados por el usuario.
- Se realizó las configuraciones para que el prototipo se conecte a una nueva red WIFI utilizando un celular.
- Se realizó el envío de datos de presion en formato tipo JSON utilizando el protocolo HTTP.

- Se implementó un sistema de recepción y envío de datos en tiempo real API REST, trabajando en conjunto con MySQL guarda los datos de presión y los resultados del preprocesamiento.

3.3 Recomendaciones

- Se recomienda realizar en una sola placa el sistema de energía y control de esa manera se ahorrará espacio y posibles interferencias electromagnéticas.
- Se recomienda seguir utilizando el chip esp32 para futuras implementaciones, sin embargo, se debe considerar cambiar del módulo para reducir el tamaño del prototipo
- Se recomienda hacer uso del protocolo WEB SOCKET que se dejó configurado para realizar una aplicación móvil para que el usuario pueda ver las mediciones y realizar la configuración de parámetros desde el móvil.
- Se recomienda realizar el preprocesamiento en el microcontrolador.
- Se recomienda seguir utilizando una pantalla con tecnología OLED, sin embargo, para agregar iconos y datos que brindan más información al usuario como hora, fecha, conectividad, etc. Es necesario aumentar su tamaño.

4. REFERENCIAS BIBLIOGRÁFICAS

- [1] B. Gal, M. Lopez, A. Martín, J. Prieto I. Barquin, "Sistema Circulatorio," Bases de la Fisiología, 2 ed, Madrid, España: Tebar , 2007, pp. 146 – 161.
- [2] D.J. Pharm, "What Is the Normal Blood Pressure Range?", Feb. 24, 2022. [Online]. Available:
https://www.medicinenet.com/blood_pressure_chart_reading_by_age/article.htm
- [3] H. Prat, G. Alarcón, I. Barquin, Hospital Clínico Universidad de Chile, "Guía para la monitorización ambulatoria de presión arterial de 24 horas", Revista Chilena de Cardiología, vol 36, dic 2017.
- [4] P.E. Cuffaro, M.S Morales, G.D. Waisman, "Monitoreo Ambulatorio de presión arterial ", Argentina, 2010, pp. 391- 393. [Online]. Available:
<https://www.saha.org.ar/pdf/libro/Cap.082.pdf>
- [5] G. Alvarez Batard, M. D. Perez Caballero, M.A. Fernández Arias, "Manual clínico del monitoreo ambulatorio de la presión arterial", Hospital Clinicoquirúrgico "Hermanos Ameijeiras", Revista Cubana de Medicina, Habana Cuba , 2006.
- [6] E. Bravo Moreno, "Dispositivos para la monitorización ambulatoria de la presión arterial MAPA", Panorama Actual del Medicamento, Chile, 2015.
- [7] A. Segui Lopez, E. Berjano Zanon, "Ingeniería inversa aplicada a un tensiómetro digital de uso doméstico", Universidad Politécnica de Valencia, Valencia – España, 2020.
- [8] M.G. Vinueza. Cahuasqui, "Diseño e implementación de un dispositivo de monitorización de presión arterial", Departamento de ingeniería eléctrica y electrónica ESPE , Ecuador, 2013.
- [9] *Procedimiento ME- 017 para la calibración de medidores de presión con salida eléctrica*, Centro Español de Metrología, España, 2020, pp. 7 - 14
- [10] A. Chandrasekhar, M. Yavarimanesh, J. Hahn, R. Mukkamala, "Formulas para explicar los algoritmos populares de estimación de la presión arterial oscilométrica", Departamento of Electrical and Computer Engineering", Michigan State University United States, November 2019.
- [11] M. Margolis. "Arduino Cookbook", OREILLY pag 404, United States of America, 2012.

- [12] F. Pezoa, J.L.Reutter, F.Suarez, M.Ugarte, "Foundations of JSON Schema", p 263, Chile, 2016.
- [13] C. Shiflett, "HTTP Developer's handbook", Sam Publishing, First ed, Indiana – EEUU, 2003
- [14] M. Masse, "REST API Design Rulebook", O'Relly, 1 ed, EEUU, 2017, p. 30.
- [15] R. Gonzales. Duque, "Phyton para todos", 1 ed, España, 2008, p. 7
- [16] M. Grinberg, "Flask Web Development", O'Relly, 1 ed, EEUU, 2014, p. 7.
- [17] R. Carrion, A. Noriega, D. Castillo. "Usando XAMPP con Bootstrap y WordPress", RasmAstur, 1 ed, España, 2015, p. 3.
- [18] S.Suehring. "MySQL BIBLIE", Wiley Publishing, 2 ed, EEUU, 2012, p. 12
- [19] R.Jurfen, "FreeCad a free extensible Cax system", EEUU, 2020, p. 4
- [20] The MathWorks, Inc, "Develop Apps Using App Designer", *The MathWorks*.
<https://www.mathworks.com/help/matlab/app-designer.html>.
- [21] Espressif, "ESP32 Series", ESP32D0WR2- V3 datasheet, 2022, p.31 [Online].
Available: https://espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

5. ANEXOS

ANEXO I. Hoja de datos 2222A.

P2N2222A					
ELECTRICAL CHARACTERISTICS ($T_A = 25^\circ\text{C}$ unless otherwise noted)					
Characteristic	Symbol	Min	Max	Unit	
OFF CHARACTERISTICS					
Collector–Emitter Breakdown Voltage ($I_C = 10\text{ mA}$, $I_B = 0$)	$V_{(BR)CEO}$	40	–	Vdc	
Collector–Base Breakdown Voltage ($I_C = 10\text{ }\mu\text{A}$, $I_E = 0$)	$V_{(BR)CBO}$	75	–	Vdc	
Emitter–Base Breakdown Voltage ($I_E = 10\text{ }\mu\text{A}$, $I_C = 0$)	$V_{(BR)EBO}$	6.0	–	Vdc	
Collector Cutoff Current ($V_{CE} = 60\text{ Vdc}$, $V_{EB(off)} = 3.0\text{ Vdc}$)	I_{CEX}	–	10	nAdc	
Collector Cutoff Current ($V_{CB} = 60\text{ Vdc}$, $I_E = 0$) ($V_{CB} = 60\text{ Vdc}$, $I_E = 0$, $T_A = 150^\circ\text{C}$)	I_{CBO}	–	0.01 10	μAdc	
Emitter Cutoff Current ($V_{EB} = 3.0\text{ Vdc}$, $I_C = 0$)	I_{EBO}	–	10	nAdc	
Collector Cutoff Current ($V_{CE} = 10\text{ V}$)	I_{CEO}	–	10	nAdc	
Base Cutoff Current ($V_{CE} = 60\text{ Vdc}$, $V_{EB(off)} = 3.0\text{ Vdc}$)	I_{BEX}	–	20	nAdc	
ON CHARACTERISTICS					
DC Current Gain ($I_C = 0.1\text{ mA}$, $V_{CE} = 10\text{ Vdc}$) ($I_C = 1.0\text{ mA}$, $V_{CE} = 10\text{ Vdc}$) ($I_C = 10\text{ mA}$, $V_{CE} = 10\text{ Vdc}$) ($I_C = 10\text{ mA}$, $V_{CE} = 10\text{ Vdc}$, $T_A = -55^\circ\text{C}$) ($I_C = 150\text{ mA}$, $V_{CE} = 10\text{ Vdc}$) (Note 1) ($I_C = 150\text{ mA}$, $V_{CE} = 1.0\text{ Vdc}$) (Note 1) ($I_C = 500\text{ mA}$, $V_{CE} = 10\text{ Vdc}$) (Note 1)	h_{FE}	35 50 75 35 100 50 40	– – – – 300 – –	–	
Collector–Emitter Saturation Voltage (Note 1) ($I_C = 150\text{ mA}$, $I_B = 15\text{ mA}$) ($I_C = 500\text{ mA}$, $I_B = 50\text{ mA}$)	$V_{CE(sat)}$	–	0.3 1.0	Vdc	
Base–Emitter Saturation Voltage (Note 1) ($I_C = 150\text{ mA}$, $I_B = 15\text{ mA}$) ($I_C = 500\text{ mA}$, $I_B = 50\text{ mA}$)	$V_{BE(sat)}$	0.6 –	1.2 2.0	Vdc	

Figura I.1 Hoja de datos 2222^a

ANEXO II. Hoja de datos 4N25

4N25 4N25A 4N26 4N27 4N28						
ELECTRICAL CHARACTERISTICS ($T_A = 25^\circ\text{C}$ unless otherwise noted) ⁽¹⁾						
Characteristic	Symbol	Min	Typ ⁽¹⁾	Max	Unit	
INPUT LED						
Forward Voltage ($I_F = 10\text{ mA}$)	V_F	–	1.15 1.3 1.05	1.5 – –	Volts	
Reverse Leakage Current ($V_R = 3\text{ V}$)	I_R	–	–	100	μA	
Capacitance ($V = 0\text{ V}$, $f = 1\text{ MHz}$)	C_J	–	18	–	pF	
OUTPUT TRANSISTOR						
Collector–Emitter Dark Current ($V_{CE} = 10\text{ V}$, $T_A = 25^\circ\text{C}$)	I_{CEO}	–	1	50 100	nA	
($V_{CE} = 10\text{ V}$, $T_A = 100^\circ\text{C}$)	I_{CEO}	–	1	–	μA	
Collector–Base Dark Current ($V_{CB} = 10\text{ V}$)	I_{CBO}	–	0.2	–	nA	
Collector–Emitter Breakdown Voltage ($I_C = 1\text{ mA}$)	$V_{(BR)CEO}$	30	45	–	Volts	
Collector–Base Breakdown Voltage ($I_C = 100\text{ }\mu\text{A}$)	$V_{(BR)CBO}$	70	100	–	Volts	
Emitter–Collector Breakdown Voltage ($I_E = 100\text{ }\mu\text{A}$)	$V_{(BR)ECO}$	7	7.8	–	Volts	
DC Current Gain ($I_C = 2\text{ mA}$, $V_{CE} = 5\text{ V}$)	h_{FE}	–	500	–	–	
Collector–Emitter Capacitance ($f = 1\text{ MHz}$, $V_{CE} = 0$)	C_{CE}	–	7	–	pF	
Collector–Base Capacitance ($f = 1\text{ MHz}$, $V_{CB} = 0$)	C_{CB}	–	19	–	pF	
Emitter–Base Capacitance ($f = 1\text{ MHz}$, $V_{EB} = 0$)	C_{EB}	–	9	–	pF	
COUPLED						
Output Collector Current ($I_F = 10\text{ mA}$, $V_{CE} = 10\text{ V}$)	I_C (CTR) ⁽²⁾	2 (20)	7 (70)	–	mA (%)	

Figura II.1 Hoja de datos 4N25

ANEXO III. Manual de Usuario

- Acceso a una nueva red WIFI

Para acceder a una nueva red WIFI el usuario debe seguir el siguiente procedimiento.

Paso 1: Energizar el dispositivo.

Paso 2: Si el usuario enciende el dispositivo y no se visualiza la pantalla de bienvenida con el numero IP que genera el dispositivo como observa en la Figura III.1.

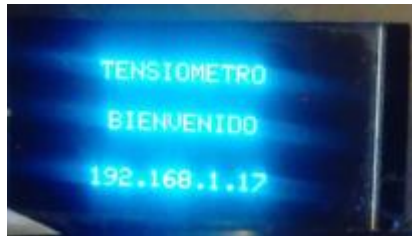


Figura III.1. Pantalla de inicio del prototipo

Quiere decir que el dispositivo no tiene guardada la clave de ninguna de las redes WiFi-cercanas, para dar acceso a una nueva red el usuario debe buscar en su teléfono celular a red AutoConnectAP, es la red generada por el dispositivo.



Figura III.2. Red AutoConnect activada

Paso 3: Elegida la red AutoConnectAP inmediatamente le saldrá la siguiente pantalla.

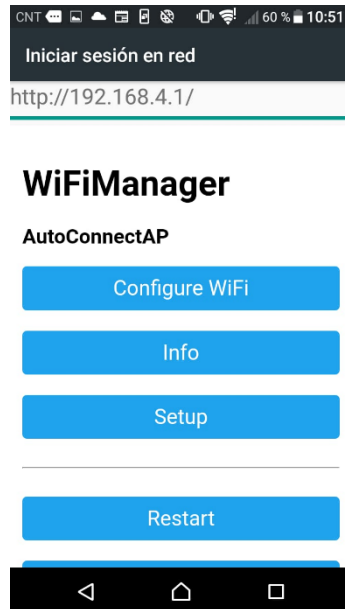


Figura III.3. Configuración WIFI

Paso 4: Se visualizan una serie de opciones, para configurar una nueva red debe ingresar a la primera opción “Configure WIFI”, inmediatamente se observa la pantalla de la Figura III.4 donde muestra las redes WIFI disponibles, se procede a escoger la red y se escribe la contraseña se espera unos 30 segundos y si las credenciales de la red WIFI estan correctas el celular y el tensiómetro se conectaran a la RED

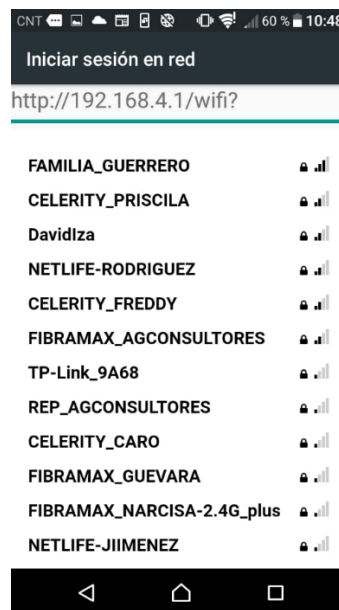


Figura III.4. Redes disponibles

- Configuraciones de Usuario

Los parámetros configurables son los siguientes:

Tiempo de estudio: Tiempo que requiere el usuario el monitoreo ambulatorio puede elegir entre 1 a 24 horas.

Inicio de sueño: Hora en la que el usuario empieza a dormir puede elegir entre 0 a 23 horas, 0 representa a las 12 am

Fin de sueño: Hora en la que el usuario termina de dormir puede elegir entre 0 a 23 horas, 0 representa a las 12 am

Tiempo entre muestras: Intervalo de tiempo en minutos entre medición de presión cuando el usuario esta despierto.

El usuario tiene 4 botones con los que puede variar los parámetros, distribuidos en el equipo como se observa la Figura III.5

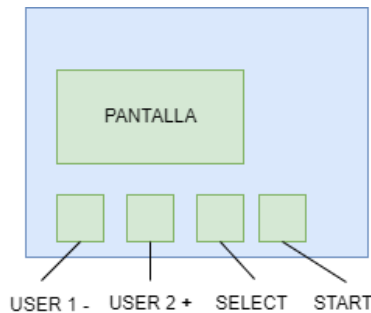


Figura III.5. Interfaz del usuario, botones y pantalla

Cuyas funciones son:

USER 1 +: EL botón permite escoger a usuario 1 y además subir el tiempo de los parámetros.

USER 2 -: EL botón permite escoger a usuario 2 y además bajar el tiempo de los parámetros.

SELECT -: EL botón permite guardar la configuración deseada, pasar a la siguiente pantalla e ingresar a configuraciones de MAPA.

START -: EL botón tiene dos funciones la primera es encender al dispositivo y la segunda es tomar la presión sin importar un MAPA previamente configurado

- Configuración de un Monitoreo de presión ambulatoria.

Paso 1: Prender al dispositivo presionado la tecla START, si existe una red WIFI previamente configurada se podrá visualizar la pantalla de la Figura III.6



Figura III.6. Pantalla de inicio

Paso 2: Se debe escoger que usuario es 1 o 2 con los botones USER 1 o USER 2 cuando el usuario haya finalizado la elección presionar SELECT

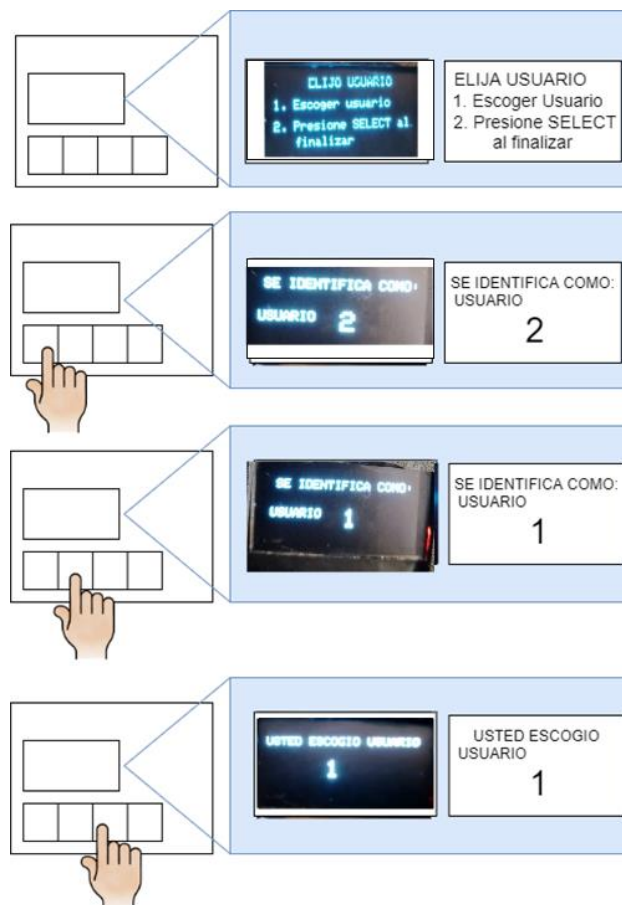


Figura III.7. Elección de usuario

Paso 3: Se visualizará la pantalla dos opciones, presionar SELECT para iniciar un MAPA.

Paso 4: Configurar los parámetros mostrados en pantalla, el orden de los parámetros es 1.

Tiempo de monitoreo

2 Inicio de sueño

3 Fin de sueño

4 Lapsos entre muestras

Con los botones USER + y USER – debe escoger el tiempo que necesite el parámetro, para guardar y pasar a la siguiente pantalla debe presionar SELECT. En la Figura III.8 se observa cómo se configura el parámetro “Tiempo de Monitoreo”.

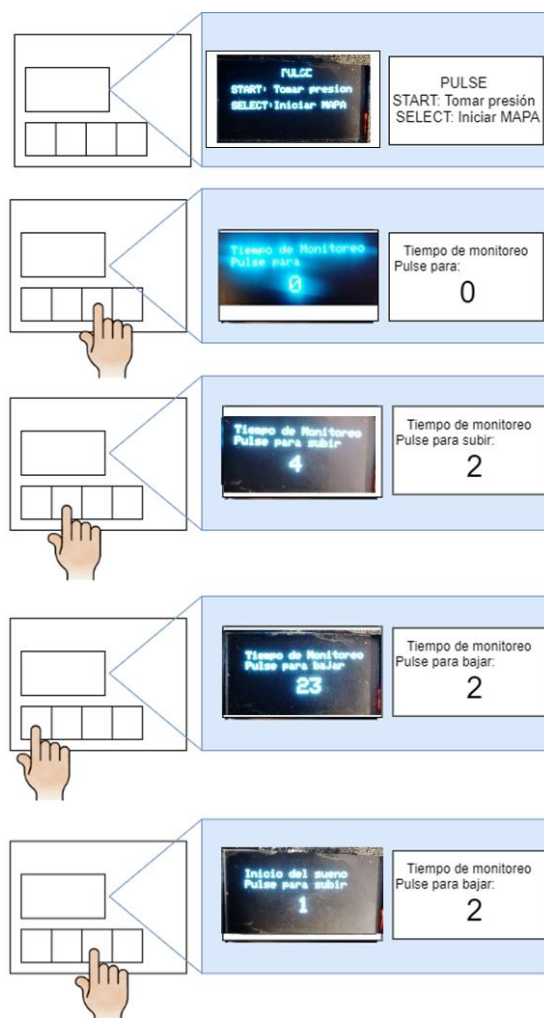


Figura III.8. Elección Tiempo de estudio

Las siguientes configuraciones como el inicio del sueño y fin de sueño seguirán la misma lógica. Con el ultimo parámetro que es Lapso entre muestras presionado SELECT se

guardan los datos en la memoria EPROM y queda configurado el monitoreo de presión arterial, se toma la primera muestra y se envía a dormir al dispositivo.

- Eliminación de un monitoreo ambulatorio de presión.

Paso 1: Presione la tecla START

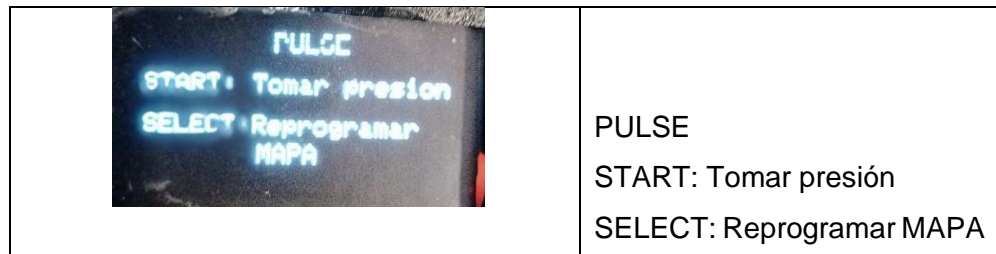


Figura III.9. Menú para eliminar MAPA

Paso 2: Coloque 0 en le primer parámetro pulse SELECT hasta siguiente hasta el ultimo parámetro, el MAPA será borrado y el dispositivo entrara en modo sueño.

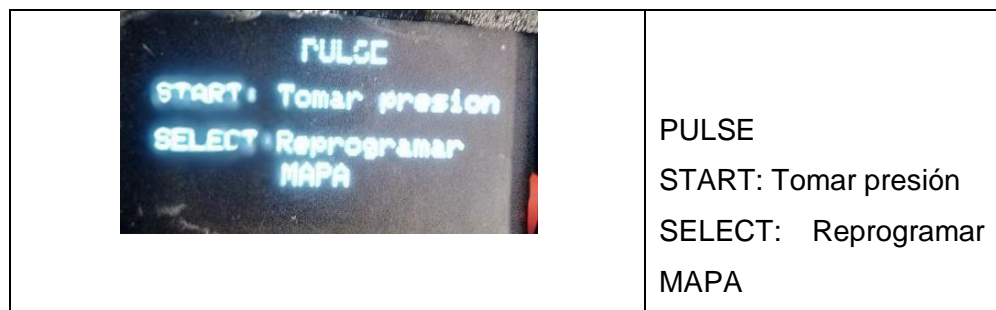


Figura III.10. Primer parámetro.

- Detección de fallas durante el muestreo de presión.

En caso de que exista una falla en el sistema del neumático durante la medición de presión se visualizar la pantalla de la Figura III.11 y el dispositivo se apagara inmediatamente.

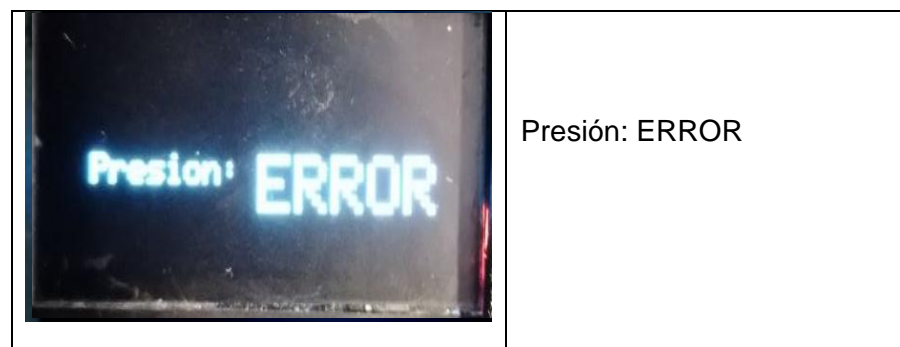


Figura III.11. Error del sistema neumático.

Si el dispositivo se mantiene por más de dos minutos sin detectar cambios se va a dormir indefinidamente hasta esperar una fuente de activación.

- Subir actualizaciones de software

Para subir una actualización de firmware el usuario debe ingresar al número de ip mostrado al inicio de la pantalla adjuntando un slash y update. Como se visualiza en la Figura III.13.



Figura III.12. IP generada por la ESP



Figura III.13. Actualización de firmware ON AIR

- Visualizar presión Sistólica, diastólica y Ritmo Cardiaco

Paso 1: Para iniciar la comunicación entre cliente y servidor iniciar Xampp y dar Start en Apache y MySQL.

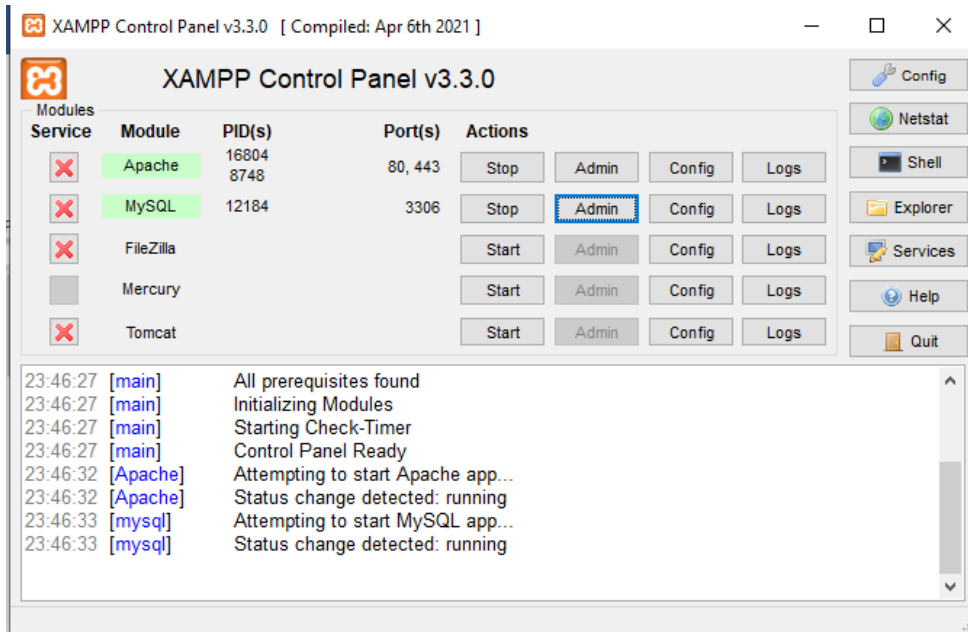


Figura III.14. Xampp

Paso 2: Se debe correr la API.

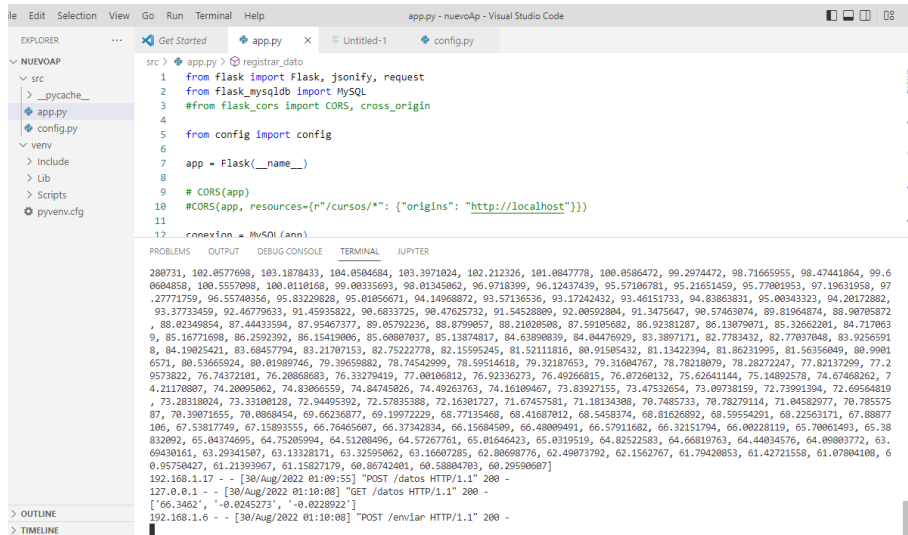


Figura III.15. API REST

Observado la terminal el usuario verificara que estan llegando los datos de presión.

Paso 3: Para realizar el preprocesamiento de la informacion. Se debe abrir la aplicación de MATLAB, pasar la página de la caratula y dar clic en el botón RUN para ver la presión

sistólica, diastólica y ritmo cardiaco como se observa en la Figura III.17



Figura III.16. Caratula de la aplicación.

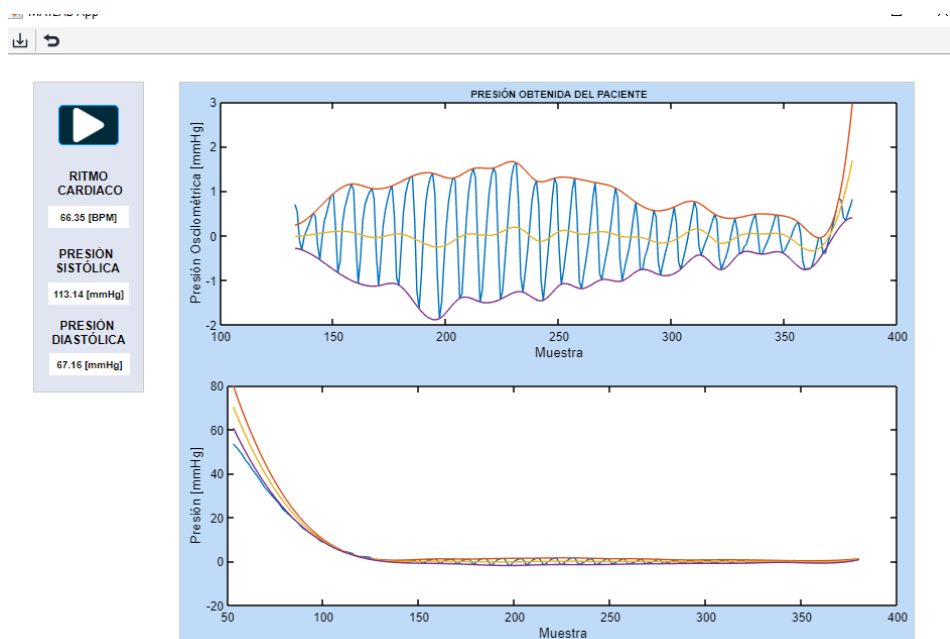


Figura III.17. OSCILOGRAMA

Para verificar los datos guardados se debe ingresar a MySQL y observar los datos

Servidor: 127.0.0.1 » Base de datos: uno002p » Tabla: datos

Examinar Estructura SQL Buscar Insertar Exportar Importar

Haga clic en la flecha desplegable para cambiar la visibilidad de la columna.

				numero	fecha9	presionalta	presionbaja
<input type="checkbox"/>	Editar			18885	2022-08-30 01:09:55	0	0
<input type="checkbox"/>	Editar	Copiar	Borrar	63.69430161	18886 2022-08-30 01:09:55	0	0
<input type="checkbox"/>	Editar	Copiar	Borrar	63.29341507	18887 2022-08-30 01:09:55	0	0
<input type="checkbox"/>	Editar	Copiar	Borrar	63.13328171	18888 2022-08-30 01:09:55	0	0
<input type="checkbox"/>	Editar	Copiar	Borrar	63.32595062	18889 2022-08-30 01:09:55	0	0
<input type="checkbox"/>	Editar	Copiar	Borrar	63.16607285	18890 2022-08-30 01:09:55	0	0
<input type="checkbox"/>	Editar	Copiar	Borrar	62.80698776	18891 2022-08-30 01:09:55	0	0
<input type="checkbox"/>	Editar	Copiar	Borrar	62.49073792	18892 2022-08-30 01:09:55	0	0
<input type="checkbox"/>	Editar	Copiar	Borrar	62.1562767	18893 2022-08-30 01:09:55	0	0
<input type="checkbox"/>	Editar	Copiar	Borrar	61.79420853	18894 2022-08-30 01:09:55	0	0
<input type="checkbox"/>	Editar	Copiar	Borrar	61.42721558	18895 2022-08-30 01:09:55	0	0
<input type="checkbox"/>	Editar	Copiar	Borrar	61.07804108	18896 2022-08-30 01:09:55	0	0
<input type="checkbox"/>	Editar	Copiar	Borrar	60.95750427	18897 2022-08-30 01:09:55	0	0
<input type="checkbox"/>	Editar	Copiar	Borrar	61.21393967	18898 2022-08-30 01:09:55	0	0
<input type="checkbox"/>	Editar	Copiar	Borrar	61.15827179	18899 2022-08-30 01:09:55	0	0
<input type="checkbox"/>	Editar	Copiar	Borrar	60.86742401	18900 2022-08-30 01:09:55	0	0

Figura III.18. Datos guardados en MySQL