



Ingeniería Electrónica

Trabajo Final de carrera:
Sistema de Monitoreo Médico - ESPill

Autores:

Nahuel García

Santiago Martín

Tutores:

Mg. Ing. Guillermo Friedrich

Ing. Adrián Laiuppa

Bahía Blanca | 6 de Marzo de 2024

Agradecimientos:

A nuestras familias por el apoyo incondicional recibido, no solo durante este proyecto sino durante toda la carrera.

A los profesores Guillermo Friedrich y Adrián Laiuppa por el apoyo permanente y su predisposición continua, con el fin de poder empezar, desarrollar y terminar este proyecto.

*Santiago Martín
Nahuel García
Bahía Blanca, 2024*

ÍNDICE

1. Resumen	8
2. Introducción	9
2.1. Solución propuesta	9
2.2. Atractivo del producto	10
3. Marco Teórico	11
3.1. Signos vitales.....	11
3.2. Temperatura corporal.....	11
3.3. Pulso cardíaco	13
3.4. Oxigenación de la sangre	14
3.5. Enfermedades asociadas al deterioro de la memoria	16
3.5.1. <i>Enfermedad de Alzheimer</i>	16
3.5.2. <i>Demencia vascular</i>	16
3.5.3. <i>Enfermedad de Parkinson</i>	16
3.6. Enfermedades respiratorias	17
3.6.1. <i>Asma</i>	17
3.6.2. <i>EPOC</i>	17
3.7. Enfermedades cardíacas	17
3.7.1. <i>Arritmias</i>	17
3.7.2. <i>Enfermedad coronaria</i>	17
3.7.3. <i>Insuficiencia cardíaca</i>	18
3.7.4. <i>Cardiomiopatías</i>	18
3.8. Enfermedades asociadas a la temperatura corporal.....	18
3.8.1. <i>Enfermedades inflamatorias</i>	18
3.8.2. <i>Enfermedades neurológicas</i>	18
4. Desarrollo del equipo	19
4.1. Descripción funcional	19
4.2. Pastillero	20
4.2.1. <i>Placa de desarrollo: ESP32 NodeMCU</i>	21
4.2.2. <i>Motor paso a paso 28BYJ-48</i>	21
4.2.3. <i>Driver ULN2003</i>	22
4.2.4 <i>Sensor de proximidad APDS-9930</i>	23
4.2.5. <i>Cargador de baterías de Litio</i>	23
4.2.6. <i>Placa del pastillero</i>	24
4.2.7 <i>Diseño 3D</i>	26
4.2.8. <i>Imágenes del pastillero</i>	27
4.3. Brazalete	28
4.3.1 <i>Medición de frecuencia cardíaca y oxigenación de la sangre</i>	28
4.3.1.1. Principio de funcionamiento	28
4.3.1.2. Principio de medición de frecuencia cardíaca	29
4.3.1.3. Principio de medición del oxígeno en sangre	29
4.3.1.4. Conexión sensor MAX30100 y ESP32	30
4.3.2. <i>Sensor de temperatura corporal</i>	31

4.3.2.1. Módulo BMP280	31
4.3.2.2. Conexión entre BMP280 y ESP32	32
4.3.3. <i>Pantalla OLED</i>	32
4.3.3.1. Conexión entre display OLED y ESP32	33
4.3.4. <i>Motor DC vibrante</i>	33
4.3.5. <i>Cargador de baterías de Litio</i>	34
4.3.6. <i>Diseño de placa del brazalete</i>	34
4.3.7. <i>Diseño 3D</i>	36
4.3.7.1. Imágenes del brazalete	37
4.4. Aplicación Móvil	37
4.4.1. <i>Interfaz Gráfica</i>	37
4.5. Servidor ThingSpeak	39
5.1. Diagramas de flujo de los Módulos	40
5.1.1. <i>Diagrama de flujo del Pastillero</i>	41
5.1.2. <i>Diagrama de flujo del Brazalete</i>	42
5.1.3. <i>Diagrama de flujo de App. Móvil</i>	43
6. Programación	44
6.1. Brazalete	44
6.1.1. <i>Función setup()</i>	44
6.1.2. <i>Función loop()</i>	45
6.2. Pastillero	47
6.2.1. <i>Función setup()</i>	47
6.2.2. <i>Función loop()</i>	48
6.3 Aplicación	51
7. Pruebas finales	55
7.1. Mediciones ESPill	55
7.2 Contraste de las mediciones	56
7.3 Cobertura del Modo AP	63
8. Conclusiones	65
9. Referencias	66

Índice de figuras

Figura 1.	Estructura y comunicación del ESPill en presencia de Wifi.	19
Figura 2.	Comunicación del sistema ESPill en ausencia de Wifi.	20
Figura 3.	ESP32 NodeMCU: Pinout placa de desarrollo del proyecto.	21
Figura 4.	Motor paso a paso 28BYJ-48 y controlador ULN2003.	22
Figura 5.	Módulo con sensor APDS-9930.	23
Figura 6.	Módulo cargador de baterías de Litio usado en pastillero y en brazalete.	24
Figura 7.	Diseño esquemático del pastillero.	24
Figura 8.	Diseño PCB del pastillero.	25
Figura 9.	Diseño 3D del pastillero en FreeCAD.	26
Figura 10.	Imágenes del pastillero.	27
Figura 11.	Sensor MAX30100 utilizado en el brazalete.	29
Figura 12.	Conexiones entre MAX30100 y ESP32 del brazalete.	30
Figura 13.	Sensor de temperatura BMP-280.	31
Figura 14.	Conexión entre sensor BMP280 y ESP32.	32
Figura 15.	Pantalla OLED para el brazalete.	33
Figura 16.	Conexión entre panel OLED y ESP32.	33
Figura 17.	Alarma vibrante en el brazalete.	34
Figura 18.	Circuito esquemático del brazalete.	35
Figura 19.	PCB del brazalete.	35
Figura 20.	Diseño 3D del brazalete.	36
Figura 21.	Diseño 3D del adaptador del oído.	36
Figura 22.	Imágenes finales del brazalete.	37
Figura 23.	Pantalla principal y pantalla de escaneo – ESPill APP.	38
Figura 24.	Pantalla de conexión y programación de compartimentos – ESPill APP.	39
Figura 25.	Credenciales para ThingSpeak - ESPill.	40
Figura 26.	Diagrama de flujo del Pastillero.	41
Figura 27.	Diagrama de flujo del Brazalete.	42
Figura 28.	Diagrama de flujo de la App. Móvil.	43
Figura 29.	Programación en bloques – petición web.	51
Figura 30.	Programación en bloques – Botón programar.	51
Figura 31.	Programación en bloques – botón aceptar.	52
Figura 32.	Programación en bloques – botón siguiente.	53
Figura 33.	Programación en bloques – botón atrás.	54
Figura 34.	Programación en bloques – botón salir.	54
Figura 35.	Equipos comerciales usados para contrastar las mediciones.	56
Figura 36.	Gráfico comparativo de temperatura corporal.	58
Figura 37.	Gráfico comparativo de pulsaciones cardiacas.	60
Figura 38.	Gráfico comparativo del oxígeno en sangre.	62
Figura 39.	Imágenes de testeo de RED representativas.	64

Índice de tablas

Tabla 1.	Rangos nominales de temperatura corporal vía oral para las distintas edades.	12
Tabla 2.	Sintomatología asociada para distintas temperaturas corporales.	12
Tabla 3.	Rangos nominales de frecuencia cardíaca para las distintas edades.	14
Tabla 4.	Sintomatología asociada a los distintos intervalos de saturación en sangre.	15
Tabla 5.	Mediciones tomadas con el brazalete colocado en el brazo del usuario.	55
Tabla 6.	Error relativo en la temperatura corporal.	57
Tabla 7.	Error relativo para las pulsaciones cardíacas.	59
Tabla 8.	Error relativo para el oxígeno en sangre.	61
Tabla 9.	Testeo de la cobertura de la red AP.	65

1. Resumen

ESPill es un dispositivo electrónico IoT que surge de la necesidad de ayudar a cuidar y a mejorar la calidad de vida de las personas mayores. La forma de llevarlo a cabo es haciendo foco en la salud y el bienestar de las personas. El sistema mide y monitorea los signos vitales del ser humano (temperatura corporal, frecuencia cardíaca y oxigenación de la sangre). Utiliza sensores no invasivos para dar alerta cuando las mediciones puedan tornarse peligrosas para la salud del usuario. El equipo también informa a la persona en qué momentos del día debe ingerir sus medicaciones.

El dispositivo se comunica inalámbricamente a través de Wifi. Para programar los horarios de ingesta de pastillas, se utiliza una aplicación móvil que emplea tecnología Bluetooth.

El sistema está diseñado para funcionar de manera autónoma mediante el suministro de baterías de corriente continua. En caso de cortes de energía eléctrica, ambos módulos seguirán comunicándose de forma inalámbrica, lo que asegura la persistencia de las alertas.

Consta de tres partes:

- El pastillero inteligente (o bloque principal) es el encargado de dispensar los medicamentos que el paciente debe ingerir. A través del giro y el movimiento de los compartimentos del sistema, dispensará al usuario las píldoras correspondientes.
- El brazalete (o bloque secundario) mide y monitorea los tres parámetros de interés mencionados. Genera eventos de alerta ante mediciones fuera de los rangos nominales.
- Aplicación móvil, permite la programación de los horarios de las pastillas que el usuario deba ingerir. También ofrece la capacidad de visualizar las tres variables medidas por el brazalete con el objetivo de que los usuarios tengan acceso a la gestión de su medicación y a la supervisión de su salud en una misma plataforma intuitiva y conveniente.

Palabras Clave

Sistemas embebidos, monitoreo remoto de pacientes, aplicaciones de IoT (internet de las cosas), telemedicina, pastilleros inteligentes, variables vitales, oximetría de pulso.

2. Introducción

La atención médica es una necesidad de vital importancia para las personas, especialmente en situaciones que requieran un monitoreo constante, ya sea debido a alguna enfermedad crónica o a la edad del paciente. En ocasiones, su estado no es crítico y no es necesario que permanezca en un centro médico bajo el cuidado de enfermeros, sino que el tratamiento puede llevarse a cabo en el hogar. Esto no solo proporciona comodidad al paciente, sino que también puede mejorar su calidad de vida al permitirle mantenerse en un entorno familiar y confortable mientras recibe el cuidado necesario.

Sin embargo, surgen desafíos cuando se necesita un monitoreo constante en el hogar, ya sea por parte de la familia o del personal médico. La falta de recursos, tiempo y la imposibilidad de garantizar una vigilancia continua pueden ser obstáculos significativos. Es en este contexto que surge la necesidad de integrar un sistema que alerte y notifique constantemente el estado del paciente.

Además de los desafíos mencionados, es importante destacar que en la actualidad se está experimentando un cambio significativo en la prestación de servicios de salud, impulsado por avances tecnológicos y cambios en las preferencias de los pacientes. Cada vez más personas buscan alternativas a los tradicionales centros médicos, optando por soluciones que les permitan recibir atención de manera remota.

En este contexto, los sistemas de monitoreo y alerta desempeñan un papel crucial al brindar una respuesta efectiva a la creciente demanda de cuidados médicos domiciliarios. Estos sistemas no solo permiten una atención personalizada y continua, sino que también contribuyen a la eficiencia de los recursos sanitarios al reducir la necesidad de hospitalización y visitas frecuentes a centros médicos.

La ingeniería electrónica y la medicina están estrechamente relacionadas en el desarrollo de tecnologías innovadoras para el monitoreo y el cuidado de la salud de las personas, incluidos los adultos mayores. A través de la aplicación de la ingeniería electrónica, se pueden crear dispositivos y sistemas avanzados que permitan un monitoreo continuo y preciso de las condiciones de salud, lo que facilita la detección temprana de problemas médicos y mejorar la calidad de vida de los pacientes.

2.1. Solución propuesta

ESPill tiene como objetivo abordar los desafíos diarios enfrentados por las personas con enfermedades crónicas o aquellos que requieren una estricta ingesta diaria de medicamentos, al mismo tiempo que monitorea el estado de salud del paciente.

Se trata de un dispositivo IoT diseñado para recordar al usuario cuando debe tomar sus medicamentos. Siendo caracterizado por su capacidad de adaptación a las necesidades

individuales de cada usuario, ESPill permite la personalización de los recordatorios según los horarios y dosis específicas de cada persona, lo que garantiza un enfoque personalizado para la gestión de la medicación.

El dispositivo está diseñado para ser fácil de configurar y utilizar. Su interfaz intuitiva y su aplicación móvil complementaria hacen que sea sencillo programar los horarios de medicación, recibir recordatorios y visualizar los datos de salud en tiempo real.

Está equipado con sensores que garantizan mediciones precisas y confiables de las variables médicas vitales (pulso cardíaco, oxigenación en sangre y temperatura corporal), proporcionando información exacta sobre el estado de salud del usuario.

2.2. Atractivo del producto

Conforme las personas envejecen, es común que enfrenten enfermedades y limitaciones que afectan su vida diaria. En muchos casos, estas condiciones requieren una atención más dedicada a la salud y, en ocasiones, asistencia externa. Por esto, un dato fundamental es el porcentaje de personas mayores en el país.

Según el Instituto Nacional de Estadísticas y Censos (INDEC, 2010), algunos datos importantes que pueden contribuir a una buena recepción en el mercado son:

- “El 14,3% es el porcentaje de la población de 60 o más años en Argentina”. Por lo cual se cree que el ESPill puede llegar a dirigirse principalmente a este porcentaje de personas.
- “El 91,7% de la población de entre 60 y 74 años ha utilizado algún bien o servicio de las TIC (Tecnologías de la Información y la Comunicación) en 2022”. Esto resulta favorable ya que quiere decir que las personas de dicha edad están cada vez más amigadas con la tecnología en general.

Otra ventaja del dispositivo es su amplio conjunto de funcionalidades a un costo más accesible en comparación con las opciones comerciales disponibles. Por ejemplo, algunas alternativas comerciales son pastilleros inteligentes que se sincronizan con dispositivos móviles a través de Bluetooth que permiten programar los horarios y las dosis de las pastillas, generando alarmas y recordatorios para cada ingesta. Una de estas opciones incluso incluye luces que indican la cantidad de pastillas a tomar y desde qué compartimento seleccionarlás, mientras que la otra cuenta con un pequeño display que muestra la hora y el estado de los recordatorios de la medicación.

Considerando este aspecto, el ESPill presenta un atractivo significativo tanto en términos de características como de precio en comparación con las opciones comerciales mencionadas.

3. Marco Teórico

3.1. Signos vitales

Los signos vitales son indicadores clave de la salud física de una persona porque proporcionan información vital sobre el funcionamiento del organismo. Estas mediciones son fundamentales en la evaluación inicial de un paciente, el monitoreo durante la atención médica y la identificación de problemas de salud (Wojcik y Kang, 2024).

Entre los signos vitales de una persona se encuentran:

- Temperatura corporal.
- Frecuencia cardíaca.
- Saturación de oxígeno en sangre.
- Presión arterial.

3.2. Temperatura corporal

El cuerpo humano está diseñado para mantener una temperatura interna constante en un rango estrecho, independientemente de las fluctuaciones ambientales. Este proceso, conocido como termorregulación, se lleva a cabo principalmente a través del sistema nervioso central y el sistema nervioso autónomo, en particular el hipotálamo, que actúa como un termostato interno del cuerpo.

Una de las unidades de medida adoptada por el SI (Sistema Internacional de unidades), es la escala en grados Celsius (o centígrados). Los lugares más comunes para medir este parámetro son:

Zona temporal: la frente es una ubicación accesible y no invasiva, lo que facilita la medición, especialmente en entornos clínicos con muchos pacientes. Suele ser ligeramente más alta que la temperatura axilar, por una diferencia de aproximadamente $0,2^{\circ}\text{C}$ a $0,5^{\circ}\text{C}$, mientras que suele ser más baja que la temperatura rectal, $0,5^{\circ}\text{C}$ o más.

Zona axilar: es un método no invasivo, especialmente en situaciones que requieren una medición rápida y cómoda. Por lo general, la temperatura axilar es de $0,3^{\circ}\text{C}$ a $0,5^{\circ}\text{C}$ más baja que la temperatura oral. Esto se debe a que la axila refleja más la temperatura superficial de la piel en lugar de la temperatura central del cuerpo. Por otro lado, la temperatura axilar tiende a ser más baja que la temperatura en la frente, con una diferencia que puede variar de $0,2^{\circ}\text{C}$ a $0,5^{\circ}\text{C}$.

Zona del oído: las mediciones se realizan con mayor precisión en el lóbulo del oído y oscilan entre $36,5^{\circ}\text{C}$ y $37,5^{\circ}\text{C}$. Esta temperatura es comparable a la temperatura oral, con una diferencia de aproximadamente $0,1^{\circ}\text{C}$ a $0,3^{\circ}\text{C}$. En situaciones prácticas, las diferencias en los valores medidos entre el lóbulo del oído y el tímpano suelen ser mínimas.

Zona rectal: es un método de medición invasivo, aunque se considera el más preciso para evaluar la temperatura central del cuerpo. Los valores típicos para la temperatura rectal oscilan entre 36,8°C y 37,5°C. La temperatura rectal suele ser más alta que la temperatura oral y temporal, con una diferencia de 0,3°C a 0,5°C.

Vía oral: generalmente se encuentra en un rango de 36,8°C a 37,2°C. Este método es conveniente y fácil de realizar, aunque es importante tener en cuenta que la ingestión reciente de alimentos o bebidas puede influir en los resultados.

Tabla 1. Rangos nominales de temperatura corporal vía oral para las distintas edades.

Rango edades	Rango temperatura corporal nominal
Recién nacidos (0 – 2 años)	35,5 – 37,5 °C
Niños (3 - 10 años)	35,5 - 37,5 °C
Niños y adultos (11 – 65 años)	36,4 - 37,6 °C
Adultos (>65 años)	35,8 – 36,9 °C

Fuente: <https://www.medicalnewstoday.com/articles/es/temperatura-normal-del-cuerpo#tabla-de-temperatura>

Tabla 2. Sintomatología asociada para distintas temperaturas corporales.

Rango de temperatura corporal	Síntomas asociados
Hipotermia (descenso de temperatura)	Extremidades frías, piel pálida, confusión, letargia, temblores, dificultad para hablar y coordinación motora comprometida. Pérdida de conocimiento en casos graves.
Febrícula (fiebre moderada)	Malestar general, fatiga, dolor de cabeza, sudoración, escalofríos y sensaciones de calor. En casos leves, la febrícula puede ser una respuesta normal del cuerpo a infecciones menores.

Fiebre alta (exceso de temperatura)	Debilidad, fatiga, dolores musculares, sudoración excesiva, pérdida de apetito, escalofríos y, a veces, delirio. La fiebre alta suele ser un signo de posibles enfermedades infecciosas.
--	--

Fuente: <https://www.healthline.com/health/es/cual-es-el-rango-normal-de-la-temperatura-corporal>

3.3. Pulso cardíaco

Otro parámetro vital en la salud y el bienestar de las personas es el pulso cardíaco, que representa la cantidad de contracciones que realiza el corazón. Se mide en pulsaciones por minuto, PPM, y el ciclo cardíaco comprende dos etapas: la sístole, que es el proceso de contracción del corazón y la diástole, que es el proceso de relajación. Entonces, un ciclo completo se refleja en una pulsación.

La frecuencia cardíaca en estado de reposo depende de factores genéticos, del estado físico, del estado psicológico, de las condiciones ambientales, de la postura, de la edad y del sexo. Esta medida se toma generalmente con la persona en reposo ya sea sentada o acostada.

En adultos, la frecuencia cardíaca en reposo generalmente oscila entre 60 y 100 latidos por minuto. Valores por encima de 100 ppm pueden indicar una posible taquicardia, que puede ser causada por estrés, fiebre, anemia o problemas cardíacos. Por otro lado, la bradicardia se presenta cuando la frecuencia cardíaca en reposo es inferior a 60 ppm. Si bien esto puede ser normal en atletas, también puede ser un indicador de problemas cardíacos. Los lugares más comunes para medir esta variable son:

Punta del dedo: colocar el instrumento medidor en la punta del dedo de la mano es común y resulta práctico ya que es una ubicación ideal debido a la abundancia de capilares sanguíneos que favorecen a las mediciones, también resulta cómodo la aplicación de los sensores en esta zona. Los valores típicos en adultos (en reposo) suelen oscilar entre 60 y 100 ppm.

Lóbulo de la oreja: puede proporcionar mediciones precisas debido a la buena perfusión sanguínea en esta área. En adultos en estado de reposo, los valores normales de frecuencia cardíaca varían entre 60 y 100 ppm.

Tabla 3. Rangos nominales de frecuencia cardíaca para las distintas edades.

Rango edad	Rango de frecuencia cardíaca nominal
Recién nacidos (0 - 1 mes)	70 – 190 ppm
Bebés (1 - 11 meses)	80 – 160 ppm
Niños (1 - 2 años)	80 – 130 ppm
Niños (3 - 4 años)	80 – 120 ppm
Niños (5 - 6 años)	75 – 115 ppm
Niños (7 - 9 años)	70 – 110 ppm
Niños (>10 años)	60 – 100 ppm
Adultos y personas mayores	60 – 100 ppm
Atletas y deportistas	40 – 60 ppm

Fuente: <https://medlineplus.gov/spanish/ency/article/003399.htm>

Los valores de la Tabla 3 representan los rangos óptimos de latidos por minuto, para distintas edades y en estado de reposo. Fuera de estos intervalos, pueden presentarse dos casos posibles: pulsaciones bajas o exceso de frecuencia cardíaca.

En los casos donde la frecuencia cardíaca es inferior al nominal, es común experimentar mareos, fatiga, debilidad, desmayos, dificultad para concentrarse. Por otro lado, cuando la frecuencia cardíaca está por encima del intervalo ideal, pueden surgir palpitaciones, mareos, debilidad, falta de aire, dolor en el pecho, entre otros síntomas.

3.4. Oxigenación de la sangre

La medición de este parámetro es fundamental en situaciones médicas donde se necesite evaluar la función respiratoria y la eficiencia en el intercambio de gases en los pulmones. Esta medida indica el nivel de oxigenación de la sangre, es decir, que tan saturada está de oxígeno. La hemoglobina es una proteína presente en los glóbulos rojos que actúa como el medio conductor para transportar el oxígeno a la sangre. Además, transporta el dióxido de carbono desde los tejidos de vuelta a los pulmones para su eliminación. Normalmente, este parámetro se mide en porcentaje de oxígeno vinculado con la hemoglobina en la sangre, conocido como oxihemoglobina.

Los valores nominales en adultos sanos generalmente se encuentran entre el 95% y el 100%. Si su valor desciende por debajo del 90%, puede indicar una disminución significativa

en la oxigenación de la sangre, lo que podría estar asociado con problemas respiratorios o cardiovasculares. La medición de este parámetro se realiza comúnmente en lugares donde se pueda acceder a los capilares sanguíneos de manera fácil y no invasiva:

Punta del dedo: esta zona resulta práctica para mediciones rápidas y no invasivas. El dedo tiene una buena perfusión sanguínea (rico en capilares sanguíneos), proporcionando resultados confiables.

Lóbulo de la oreja: esta zona tiene una buena perfusión sanguínea y puede ser una buena alternativa cuando la medición en el dedo no es posible o conveniente.

Tanto la punta del dedo como el lóbulo de la oreja ofrecen rangos típicos de SpO₂ similares, que suelen ser de 95% a 100%.

Tabla 4. Sintomatología asociada a los distintos intervalos de saturación en sangre.

Intervalo SpO ₂	Estado de salud	Síntomas asociados
95% a 100% (aplica para todas las edades)	Óptimo.	Saludable, hemoglobina saturada con oxígeno.
90% a 94%	Subóptimo.	Enfermedad pulmonar obstructiva crónica (EPOC), insuficiencia cardíaca congestiva o problemas respiratorios.
Menor a 90%	Hipoxemia.	Neumonía, insuficiencia respiratoria, insuficiencia cardíaca aguda, shock u otras condiciones médicas graves.
Menor a 85%	Hipoxemia severa (riesgo severo de salud).	Insuficiencia respiratoria aguda, paro cardíaco, traumatismo grave u otras emergencias médicas.

Fuente: <https://www.neumotec.com/cuales-son-los-niveles-normales-de-saturacion-de-oxigeno-en-sangre/>

3.5. Enfermedades asociadas al deterioro de la memoria

3.5.1. Enfermedad de Alzheimer

Es una afección neurodegenerativa y progresiva que afecta principalmente a adultos mayores, causando deterioro cognitivo y funcional, según los especialistas de Alzheimer.gov (2024). Sus síntomas más comunes son problemas en la memoria debido a su deterioro, dificultades para realizar tareas cotidianas, desorientación en tiempo-espacio, pérdida de juicio, cambios en el comportamiento y el estado de ánimo, así como dificultades cognitivas más amplias, entre otros.

En cuanto a las causas, se sabe que la enfermedad de Alzheimer está influenciada por una combinación de factores genéticos y del estilo de vida que la persona ha tenido a lo largo de los años. La predisposición genética desempeña un papel importante, y las personas con antecedentes familiares de la enfermedad tienen un mayor riesgo.

El acompañamiento y la asistencia de terceros son fundamentales en el manejo de esta enfermedad. Es crucial asegurarse de que el paciente reciba la atención médica adecuada y siga el tratamiento recetado por su médico, así como mantenerlo mentalmente activo con actividades estimulantes como rompecabezas, juegos de memoria y lectura.

En este contexto, la necesidad de administrar las medicaciones recetadas en los horarios indicados por el médico cobra una gran importancia. El apoyo de los cuidadores y familiares garantiza que el paciente reciba el tratamiento adecuado en el momento oportuno, lo que contribuye a mantener su salud y calidad de vida en la medida de lo posible.

3.5.2. Demencia vascular

La enfermedad o demencia vascular cerebral ocurre cuando hay daños en los vasos sanguíneos del cerebro, lo cual puede ser causado por accidentes cerebrovasculares, enfermedades arteriales periféricas, hipertensión o diabetes.

La demencia vascular puede producir síntomas similares a los de la enfermedad del Alzheimer, como pérdida de memoria y dificultades cognitivas.

3.5.3. Enfermedad de Parkinson

Si bien esta enfermedad se asocia principalmente con síntomas motores como temblores, rigidez y dificultad para caminar, la enfermedad de Parkinson también puede provocar cambios cognitivos, incluida la pérdida de memoria y otros síntomas relacionados con la demencia (Wojcik y Kang, 2024).

3.6. Enfermedades respiratorias

3.6.1. Asma

Es una enfermedad crónica del sistema respiratorio que afecta los conductos bronquiales, encargados de transportar el aire hacia y desde los pulmones. Esta condición causa inflamación y estrechamiento de las vías respiratorias, lo que provoca síntomas como dificultad para respirar, tos y opresión en el pecho.

Los pacientes con asma que experimentan una disminución significativa en la saturación de oxígeno pueden necesitar oxígeno suplementario y otros tratamientos para mejorar la función respiratoria y prevenir complicaciones graves. Por lo tanto, la monitorización continua de la saturación de oxígeno en sangre es fundamental para el bienestar de los pacientes con asma.

3.6.2. EPOC

Llamada enfermedad pulmonar obstructiva crónica, se caracteriza por la obstrucción del flujo de aire en los pulmones. Es progresiva y está asociada generalmente al tabaquismo. Dentro de esta enfermedad se encuentra la bronquitis crónica, que implica una inflamación de las vías respiratorias, y el enfisema, que es el daño en los alvéolos de los pulmones.

3.7. Enfermedades cardíacas

3.7.1. Arritmias

Son trastornos del ritmo cardíaco que pueden provocar una frecuencia cardíaca alta (taquicardia) o baja (bradicardia). Algunos ejemplos son la fibrilación auricular, la taquicardia ventricular, bradicardia sinusal y el bloqueo cardíaco. Estas condiciones pueden ser causadas por problemas en el sistema eléctrico del corazón, daño en el músculo cardíaco, trastornos metabólicos, consumo de ciertos medicamentos o factores genéticos.

3.7.2. Enfermedad coronaria

Es una condición en la que las arterias que suministran sangre al corazón se estrechan o se bloquean, reduciendo el flujo sanguíneo hacia el músculo cardíaco. En algunos casos, una obstrucción significativa en las arterias coronarias puede provocar una disminución de la frecuencia cardíaca.

3.7.3. Insuficiencia cardíaca

Es una afección en la que el corazón no puede bombear suficiente sangre para satisfacer las necesidades del cuerpo. En etapas avanzadas de la enfermedad, el corazón puede volverse menos eficiente y la frecuencia cardíaca, en reposo, disminuye como resultado de la insuficiencia cardíaca y la respuesta compensatoria del sistema nervioso.

3.7.4. Cardiomiopatías

En ocasiones estas enfermedades pueden provocar que el músculo cardíaco se vuelve anormalmente grueso, generando arritmias por la frecuencia elevada. En otros casos, el músculo cardíaco se debilita, lo que puede resultar en arritmias debido a la disminución de la frecuencia cardíaca.

3.8. Enfermedades asociadas a la temperatura corporal

3.8.1. Enfermedades inflamatorias

La inflamación en el cuerpo produce un aumento de la temperatura corporal. En ocasiones, la fiebre es un signo de actividad inflamatoria aguda y puede acompañar a otros síntomas característicos.

3.8.2. Enfermedades neurológicas

Estas enfermedades pueden interferir la capacidad del cerebro para regular la temperatura corporal, provocando una disminución de la temperatura. Algunos ejemplos de este tipo de enfermedades incluyen el accidente cerebrovascular (ACV), la esclerosis múltiple y la enfermedad de Parkinson.

El ACV es una condición que ocurre cuando el suministro de sangre del cerebro se interrumpe repentinamente, ya sea debido a un bloqueo de una arteria cerebral (accidente cerebrovascular isquémico) o por la ruptura de un vaso sanguíneo en el cerebro (accidente cerebrovascular hemorrágico). El ACV puede afectar la capacidad del hipotálamo, la parte del cerebro responsable de mantener la temperatura corporal estable. Esto puede resultar en una disminución de la temperatura corporal, especialmente en casos de daño cerebral extenso que afecta al hipotálamo.

4. Desarrollo del equipo

4.1. Descripción funcional

El sistema se compone de dos elementos principales: el brazalete y el pastillero. El primero, se encarga principalmente de medir las variables vitales del paciente y alertar a terceros ante anomalías en éstas. Por otro lado, el pastillero es el que entrega las píldoras en los horarios previamente programados mediante la aplicación móvil.

Ambos están diseñados para complementarse facilitando el intercambio de datos del paciente en tiempo real. El sistema requiere estar dentro del alcance de una red Wifi para garantizar la comunicación efectiva. En caso de interrupción en la red, el pastillero cambiará su conexión a modo Access Point (AP) para mantener la transferencia de datos con el brazalete. A continuación, se presenta el diagrama funcional del sistema, con el pastillero como dispositivo central:

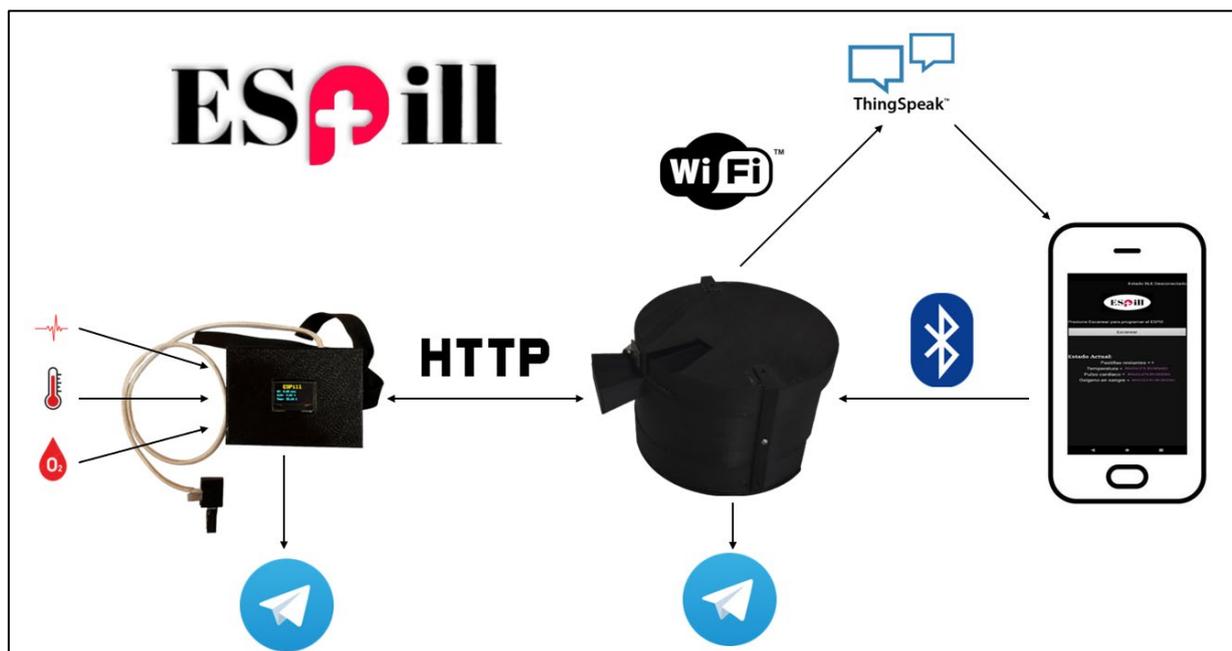


Figura 1. Estructura y comunicación del ESPill en presencia de Wifi.

Cuando la red Wifi es interrumpida, el brazalete continuará configurado como estación, pero el pastillero se reconfigurará al modo AP, generando una red llamada "Pastillero", aceptando la conexión de su par para continuar la comunicación.

Esta conexión se realiza utilizando el protocolo HTTP, el cual corre bajo la capa de transporte TCP/IP. A continuación, se muestra el diagrama de conexión en ausencia de Wifi.

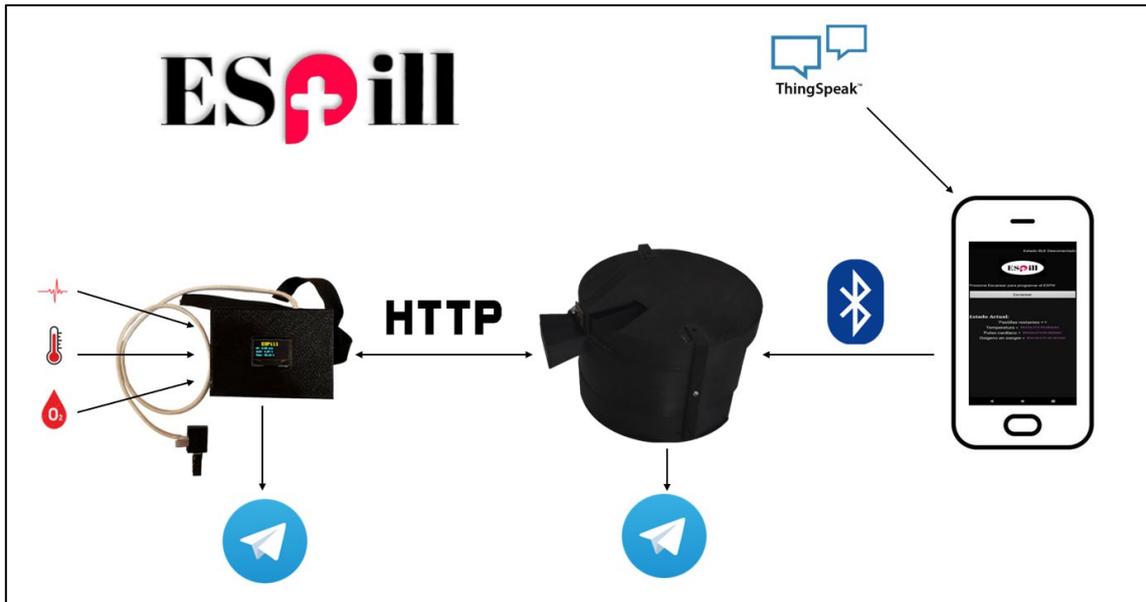


Figura 2. Comunicación del sistema ESPill en ausencia de Wifi.

4.2. Pastillero

Es el encargado de gestionar los horarios en los que el paciente debe tomar sus medicaciones, así como de transmitir los datos recopilados a la nube. Este dispositivo está formado por compartimentos donde el usuario ingresa sus medicamentos. Además, está equipado con una alarma sonora y visual para notificar al paciente (y a terceros) la ingesta. Para garantizar su funcionamiento continuo, cuenta con un banco de baterías de litio en caso de interrupción del suministro eléctrico.

El hardware incluye:

- ESP32 NodeMCU.
- Motor paso a paso 28BYJ-48 con driver ULN2003.
- Sensor APDS-9930 (detector de proximidad humana).
- Módulo cargador de baterías TP4056.
- Baterías de Litio 18650 de 3,7V.
- Fuente DC/DC elevadora de tensión.
- Fuente 220 VAC a 5 VDC por 2A.
- Botón RST externo.
- Buzzer para alarma.
- Fusible para protección del equipo.

4.2.1. Placa de desarrollo: ESP32 NodeMCU

Es un microcontrolador de 32-bit del fabricante Espressif Systems, de bajo costo, de gran potencia y de bajo consumo. Posee comunicación Wifi, Bluetooth y es ideal para aplicaciones IoT. Entre sus principales características se encuentran:

- Procesador de doble núcleo Soc. (System on a chip) ESP32-D0WDQ6, que integra un Tensilica Xtensa LX6 a una frecuencia máxima de 240 MHz.
- Interfaces de comunicación I²C, SPI y UART.
- Memoria Flash integrada de 4 MB.
- Conectividad Wifi a través de IEEE 802.11 b/g/n.
- Bluetooth de bajo consumo (BLE).
- ADC de 12-bit.
- Consumo en reposo de 20 a 30 mA.

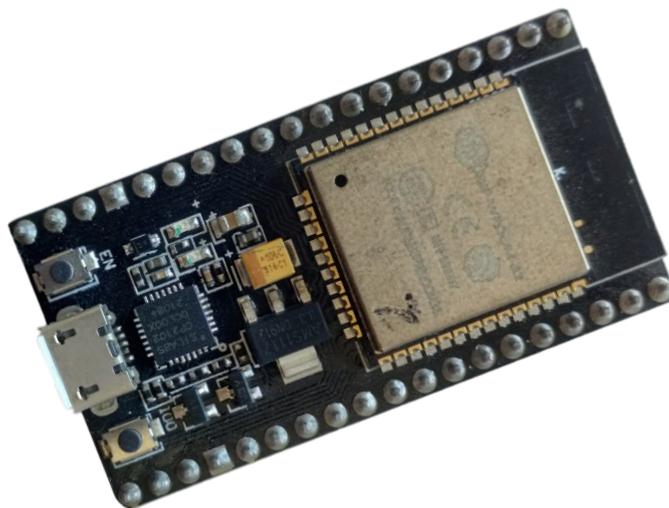


Figura 3. ESP32 NodeMCU: Pinout placa de desarrollo del proyecto.

4.2.2. Motor paso a paso 28BYJ-48

Para girar el pastillero se necesita un motor paso a paso porque ofrece una gran precisión y control en sus movimientos. Las principales características de este motor son:

- Voltaje nominal: 5 a 12 VDC.
- Corriente nominal: 50 mA por fase, aproximadamente.
- Cantidad de fases: 4 (son dos bobinas con punto medio).
- Tipo de motor: paso a paso unipolar.
- Ángulo de paso: 5,625 grados por paso (64 pasos por revolución).

- Reducción de engranajes: relación 64:1.
- Tamaño: compacto y muy preciso para generar los pasos.

Sin embargo, el microcontrolador no puede entregar la corriente necesaria para controlar el motor, ya que su capacidad de corriente no es suficiente. Por ende, es necesario hacer uso de un controlador de corriente.

4.2.3. Driver ULN2003

Es fundamental emplear un controlador de motor que funcione como un amplificador de corriente para controlar el motor paso a paso. El driver elegido es el ULN2003 cuyas características son:

- Varias líneas de control, ya que cuenta con hasta 7 salidas independientes.
- Transistores en configuración Darlington: dos transistores bipolares en cascada que producen ganancia en corriente. Esto se replica para cada salida del controlador.
- Corriente de salida: cada salida puede manejar cargas de hasta 500 mA.
- Protecciones: cada salida posee un diodo de protección para que el comportamiento inductivo del motor (cuando se desenergiza) no sobrecargue las etapas Darlington.

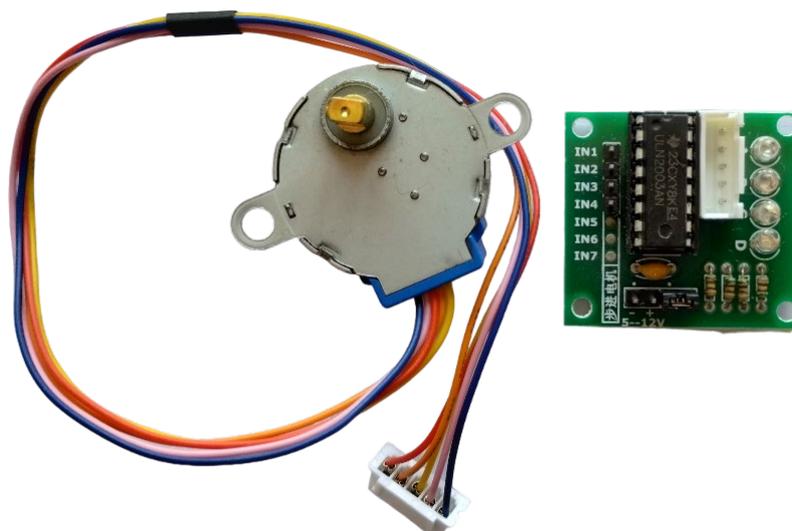


Figura 4. Motor paso a paso 28BYJ-48 y controlador ULN2003.

4.2.4 Sensor de proximidad APDS-9930

La función del sensor de proximidad APDS-9930 es la de detectar el movimiento de la persona al retirar las píldoras del pastillero. Para lograr esto, emite una luz infrarroja y luego recibe la luz reflejada. Sus principales características son:

- Voltaje de alimentación: 3,3 VDC.
- Bajo consumo eléctrico: consumo de corriente de alrededor de 100 μ A.
- Interfaz de comunicación: utiliza I²C para poder enviar a un microcontrolador el estado del sensor.
- Detección configurable: puede detectar objetos a distancias de hasta varios centímetros, dependiendo de la inicialización que se haga de los parámetros por software (sensibilidad, muestreo de la luz ambiental, modo de operación, etc.).
- Filtro de luz ambiente: con el fin de mejorar las mediciones en diferentes condiciones de iluminación.
- Pequeño y compacto: mide 10 mm de largo, 8 mm de ancho y 1.5 mm de alto, aproximadamente.



Figura 5. Módulo con sensor APDS-9930.

4.2.5. Cargador de baterías de Litio

Se emplean dos módulos TP4056: uno para el pastillero y otro para el brazalete. Se trata de un circuito integrado diseñado para la carga de baterías de litio-ion y litio-polímero.

Este módulo proporciona una solución de carga segura y eficiente para baterías de litio, con facilidad de uso y bajo costo. Algunas de sus características son:

- Tensión de alimentación: 5 VDC.
- Corriente de carga: ajustable mediante la selección de resistencias externas conectadas al módulo. Por defecto viene seteado para cargar a 1A como máximo.
- Protección contra sobrecarga: para evitar daños a la batería durante el proceso de carga. Por encima de aproximadamente 4,2V el integrado desconectará la batería.
- Protección contra sobredescarga: para evitar que la batería se descargue completamente, de modo de cuidar la capacidad de la pila.

Una vez finalizado y verificado el diseño esquemático, se realizó el diseño del PCB. Se trata de una placa de una sola cara con plano a tierra, como se aprecia en la Figura 8.

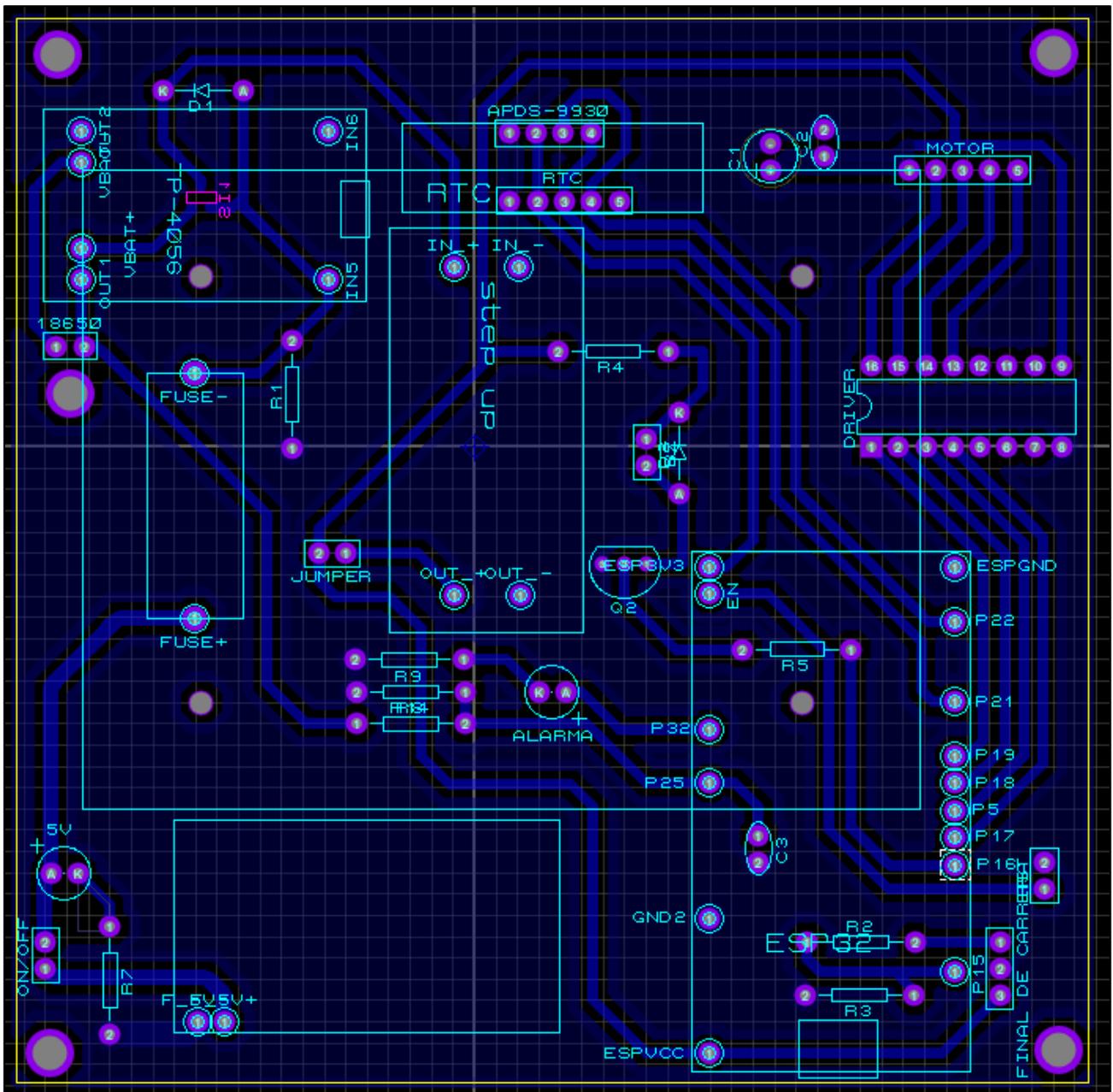


Figura 8. Diseño PCB del pastillero.

Se utiliza una fuente conmutada de 220 VAC a 5 VDC y se emplea un MOSFET canal N que actúa como interruptor entre las dos tensiones disponibles provenientes de la fuente y de la batería. Logrando cargar las baterías a la máxima corriente permitida por el TP4056.

4.2.7 Diseño 3D

El dispositivo debe ser cómodo para el usuario y necesita tener la menor cantidad de partes móviles para descartar cualquier tipo de error mecánico, porque tiene un único movimiento para dispensar pastillas. A continuación, se detallan los diseños de las carcasas del pastillero realizadas en FreeCAD:

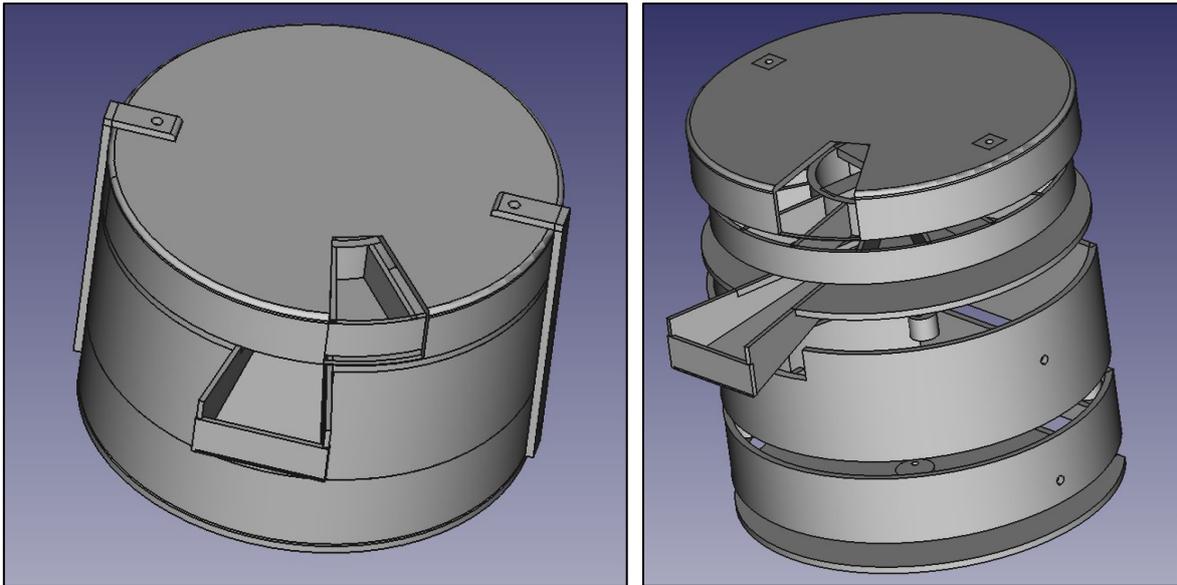


Figura 9. Diseño 3D del pastillero en FreeCAD.

4.2.8. Imágenes del pastillero



Figura 10. Imágenes del pastillero.

Como se ve en las imágenes, el módulo consta de varias capas para lograr un mejor armado. Sus medidas son de 11 cm de alto por 23 cm de diámetro.

4.3. Brazalete

Entre sus funciones principales se encarga de monitorear los signos vitales del paciente: temperatura corporal, frecuencia cardiaca y oxigenación de la sangre; y alertar a terceros acerca de anomalías detectadas. Luego de ser procesadas, se envían al pastillero para poder almacenarlas en la nube. Está compuesto por:

- ESP32 NodeMCU.
- Sensor MAX30100 (pulsaciones cardiacas y saturación de oxígeno).
- Sensor BMP280 (temperatura corporal).
- Pantalla OLED 0,96 pulgadas.
- Módulo cargador de baterías TP4056.
- Batería Litio de 3,7V y 2200 mAh.
- Fuente 220V AC / 5V DC
- Fuente DC/DC elevadora de tensión.
- Motor vibrante.
- Botón RST externo.

4.3.1 Medición de frecuencia cardiaca y oxigenación de la sangre

El sensor MAX30100 (Robocraze, 2023) es un sensor óptico del fabricante Maxim Integrated, que funciona según el comportamiento que la sangre tiene ante la luz, en función de su grado de saturación de oxígeno. Para esto, incorpora dos leds, uno de espectro rojo y otro de infrarrojo.

4.3.1.1. Principio de funcionamiento

Obtiene sus lecturas emitiendo dos longitudes de onda distintas (660 nm la roja y 940 nm la infrarroja), provenientes de los dos leds. Mide la absorbancia de la sangre pulsante a través de un fotodetector (fotopletoisgrafía) y es lo que en definitiva permite al sensor calcular la frecuencia cardíaca y la saturación de oxígeno. Cuanto mejor sea la perfusión de la sangre, más precisa será la medición que tome el equipo. Sus características principales son:

- Medición mediante PPG para medir la absorción de luz en la sangre.
- Detección de pulso óptica.
- Tasa de muestreo programable y corriente de led variable, para ahorro de energía.
- Resolución del ADC configurable de 14 a 16 bit.
- Interfaz I²C para comunicación con microcontroladores.
- Bajo consumo de energía.
- Amplio rango de operación.
- Filtrados de cancelación del ruido ambiental.
- Tamaño compacto.



Figura 11. Sensor MAX30100 utilizado en el brazalete.

4.3.1.2. Principio de medición de frecuencia cardíaca

La hemoglobina oxigenada (HbO₂) de la sangre arterial tiene la característica de absorber la luz IR porque cuanto más roja es la sangre, mayor es la hemoglobina y mayor es la luz IR se absorbida. A medida que la sangre circula a través de los capilares sanguíneos con cada latido del corazón, la cantidad de luz reflejada cambia, creando una forma de onda cambiante a la salida del fotodetector.

4.3.1.3. Principio de medición del oxígeno en sangre

La hemoglobina presenta diferencias significativas en su absorción de luz según su estado de oxigenación. Estas diferencias se pueden describir en términos de la absorción de distintas frecuencias de luz:

- 1) Espectro visible: la hemoglobina oxigenada y desoxigenada tienen patrones distintos de absorción de luz en el espectro visible. La HbO₂ absorbe menos luz roja (de 600-750 nm) y más el espectro azul-verde (de 400-450 nm). Por otro lado, la Hb absorbe más luz en la región del rojo y menos en el azul-verde. Este comportamiento se traduce en diferencias de color: la sangre rica en oxígeno se ve roja brillante debido a la menor absorción de la luz roja, mientras que la sangre pobre en oxígeno tiene un tono más oscuro o azulado.
- 2) Infrarrojo cercano (NIR): en este rango, entre 700 y 1000 nm, las diferencias de absorción entre la Hb y la HbO₂ disminuyen, pero son significativas para la detección y medición. La hemoglobina oxigenada absorbe menos en este rango que la desoxigenada, lo que permite técnicas como la oximetría de pulso, que miden la saturación de oxígeno en sangre de manera no invasiva utilizando longitudes de onda específicas dentro del espectro visible e infrarrojo cercano.

4.3.1.4. Conexión sensor MAX30100 y ESP32

La conexión entre el microcontrolador y el sensor se realiza a través del protocolo I²C. Este estándar simplifica la comunicación entre ambos dispositivos, utilizando solo dos líneas de señal y una tierra o masa compartida. Facilita el intercambio de información a velocidades de hasta 3,4MHz.

La metodología de comunicación de este bus es serial y síncrona. Una de las señales, genera los pulsos de reloj para sincronizar el sistema, mientras que la otra se utiliza para la transferencia de datos. Las señales que utiliza este protocolo son:

- SCL (System Clock): línea de pulsos de reloj que sincroniza el sistema.
- SDA (System Data): línea de transmisión de datos.
- GND (Masa): tierra común de todos los dispositivos conectados al bus.

SDA y SCL utilizan una configuración de drenaje abierto, por lo que están asociadas a transistores de efecto de campo (FET). Estas líneas deben ser polarizadas en estado alto (conectadas a alimentación mediante resistencias de “pull-up”), lo que establece una estructura que permite conectar múltiples dispositivos en paralelo.

A continuación, se muestra la conexión de los puertos I²C del microcontrolador y el sensor:

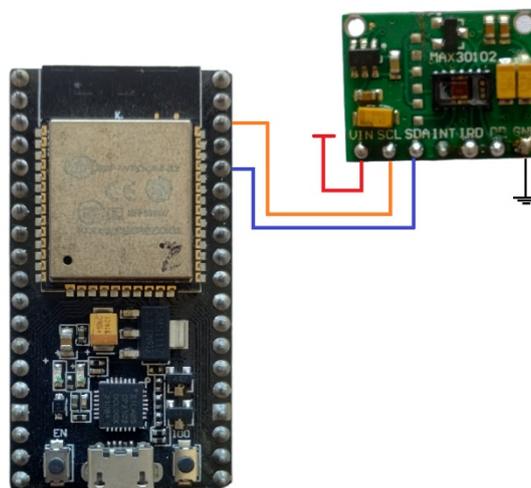


Figura 12. Conexiones entre MAX30100 y ESP32 del brazalete.

4.3.2. Sensor de temperatura corporal

4.3.2.1. Módulo BMP280

Este módulo tiene un sensor de temperatura y presión atmosférica, basado en la tecnología piezo-resistiva del fabricante BOSCH, con robustez a interferencias electromagnéticas, alta precisión y linealidad. Se conecta al microcontrolador a través de una interfaz I²C o SPI (Gras, 2020; Llamas, 2024).

La elección de este sensor se debe a su alta sensibilidad en las mediciones, capaz de detectar cambios pequeños en la temperatura corporal de las personas. Entre sus principales características se encuentran:

- Mide temperatura y presión atmosférica.
- Interfaz de comunicación digital configurable (I²C o SPI).
- Amplio rango de temperatura y presión: mide temperaturas de -40 °C a 85 °C y presiones desde 300 hasta 1100 hPa.
- Bajo consumo de energía.
- Compensación por temperatura: cuenta con un sensor de temperatura interno para compensar las variaciones térmicas.
- Modos de operación estándar, de bajo consumo y de alta resolución.
- Tamaño compacto.
- Calibración integrada.

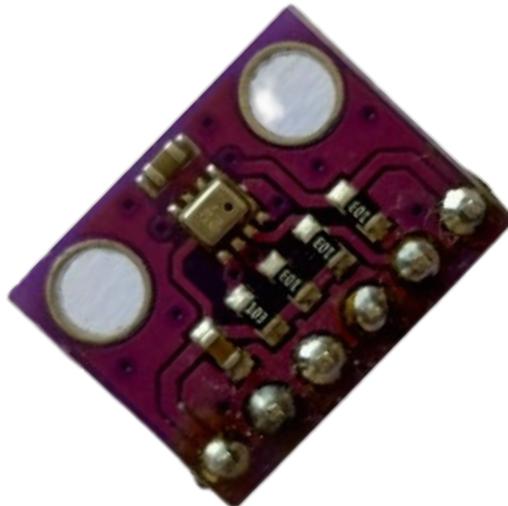


Figura 13. Sensor de temperatura BMP-280.

4.3.2.2. Conexión entre BMP280 y ESP32

Para la transmisión de datos, este dispositivo cuenta con el bus I²C, y cuenta con las resistencias de pull up integradas lo que facilita su manejo. El diagrama de conexión que utiliza este protocolo es:

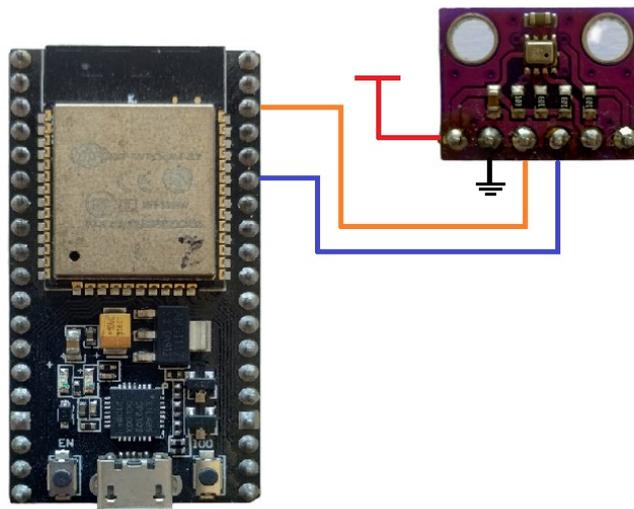


Figura 14. Conexión entre sensor BMP280 y ESP32.

4.3.3. Pantalla OLED

El brazalete posee un panel OLED para visualizar las mediciones. Sus características principales son:

- Pantalla OLED de 0,96 pulgadas. Resolución 128x64 pixeles.
- Alimentación 3 VDC a 5 VDC.
- Controlador: SSD1306, interfaz de comunicación I²C.
- Tamaño pequeño: 26,7 mm por 19,26 mm por 1,85 mm.

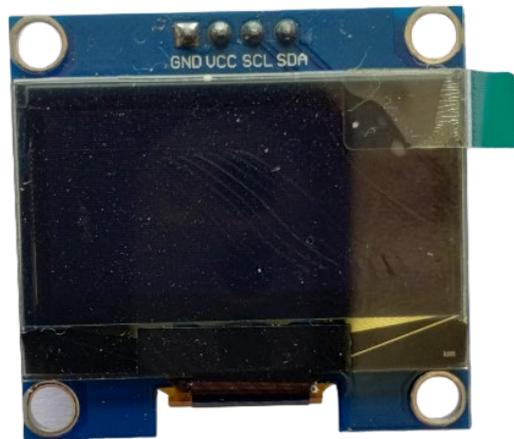


Figura 15. Pantalla OLED para el brazalete.

4.3.3.1. Conexión entre display OLED y ESP32

Para que funcione el display, las conexiones para el bus I²C son:

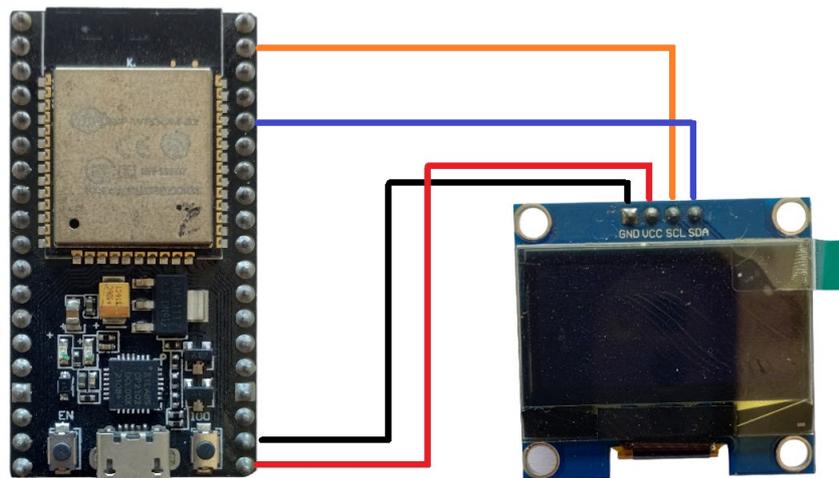


Figura 16. Conexión entre panel OLED y ESP32.

4.3.4. Motor DC vibrante

Utilizado en el sistema del brazalete y su función es alertar acerca de mediciones peligrosas registradas por el equipo. Tiene un rotor desequilibrado y al aplicarle energía eléctrica, el rotor gira de manera excéntrica, provocando vibraciones mecánicas sobre su eje. Sus características principales son:

- Alimentación: 3 a 5 VDC.
- Consumo de 100mA a 3,3 VDC.
- Tamaño compacto: 12 mm de alto, 6 mm de diámetro y un peso de 20 gr.

Para su implementación, es necesario una etapa a transistor BJT para generar la conmutación a la corriente de funcionamiento del motor. El diodo en paralelo con la carga es de protección del motor.



Figura 17. Alarma vibrante en el brazalete.

4.3.5. Cargador de baterías de Litio

La batería entrega una tensión constante al sistema y la fuente elevadora convierte los 3,7V en 5V. Los sensores y el motor de vibración se alimentan a través de la ESP32 a 3,3V. El circuito del cargador de baterías es el mismo que para el caso del pastillero.

4.3.6. Diseño de placa del brazalete

A continuación, se puede ver el circuito esquemático del brazalete realizado:

4.3.7. Diseño 3D

Este dispositivo debe resultar cómodo de llevar y lo suficientemente pequeño para que no sea evidente. Por este motivo se decidió comprimir dentro de lo posible la electrónica interna dentro del chasis, y se eligió un adaptador de oído tipo broche, para que los sensores puedan ejercer una presión estable en el lóbulo del paciente. A continuación, se encuentran los diseños 3D realizados:

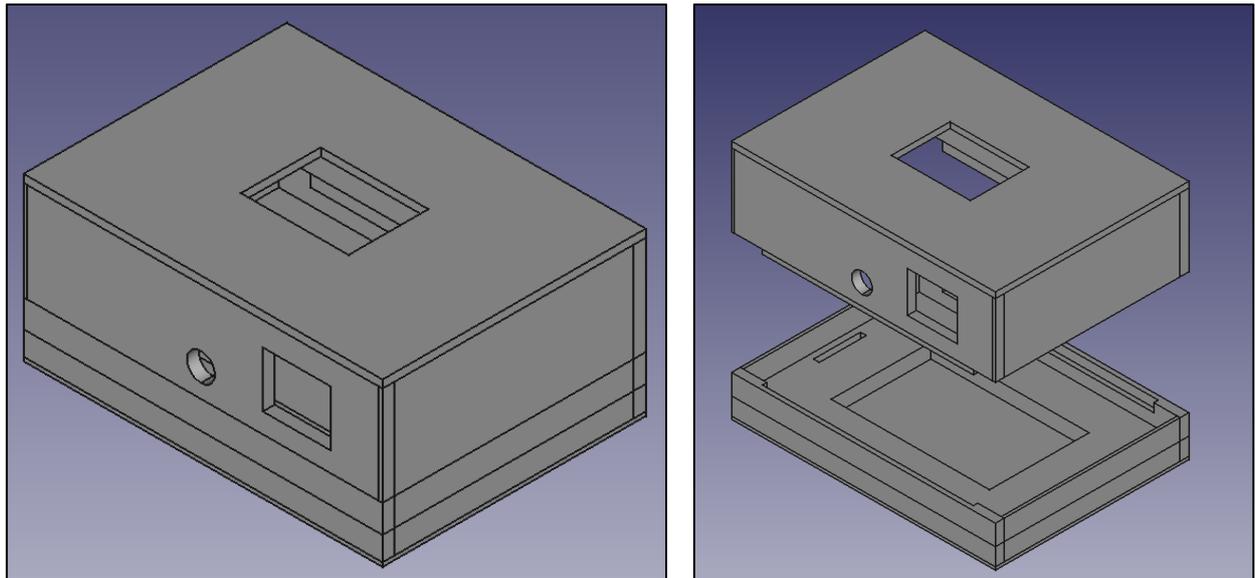


Figura 20. Diseño 3D del brazalete.

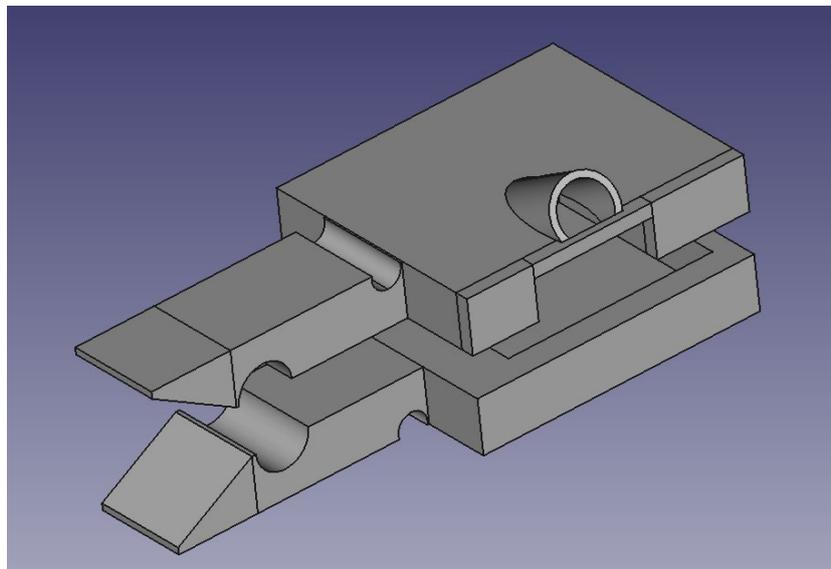


Figura 21. Diseño 3D del adaptador del oído.

4.3.7.1. Imágenes del brazalete

Se trata de la carcasa 3D y un cable de 4 conductores dirigido hacia el adaptador de censado. En la parte inferior de la carcasa se encuentra la batería plana de litio. En la Figura 22 se muestra el diseño final:

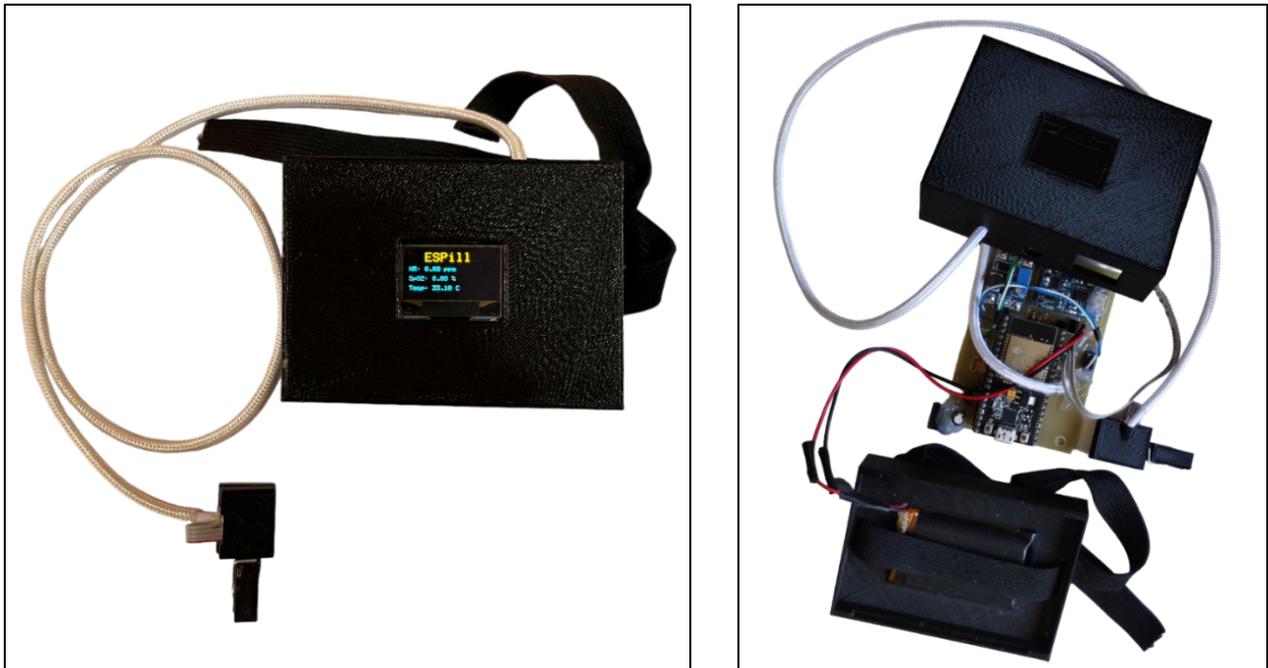


Figura 22. Imágenes finales del brazalete.

4.4. Aplicación Móvil

A través de Bluetooth se permite programar el pastillero con los horarios que se deben ingerir las medicaciones y desde el dispositivo móvil será posible ver las mediciones del sistema.

4.4.1. Interfaz Gráfica

Cuando se inicia la aplicación, se visualiza el estado actual de las variables contenidas en la nube. Estos parámetros se actualizan cada cinco segundos y solo se debe tener conexión a internet. En esta pantalla, tendremos la posibilidad de escanear los dispositivos que se encuentren cercanos, y para esto se deben aceptar los permisos para que la aplicación pueda utilizar el Bluetooth (Concha, 2015).

Luego de establecer los permisos, el sistema comienza el escaneo de dispositivos Bluetooth cercanos hasta encontrar el dispositivo llamado ESPill_BLE, como se indica en la segunda imagen a continuación.

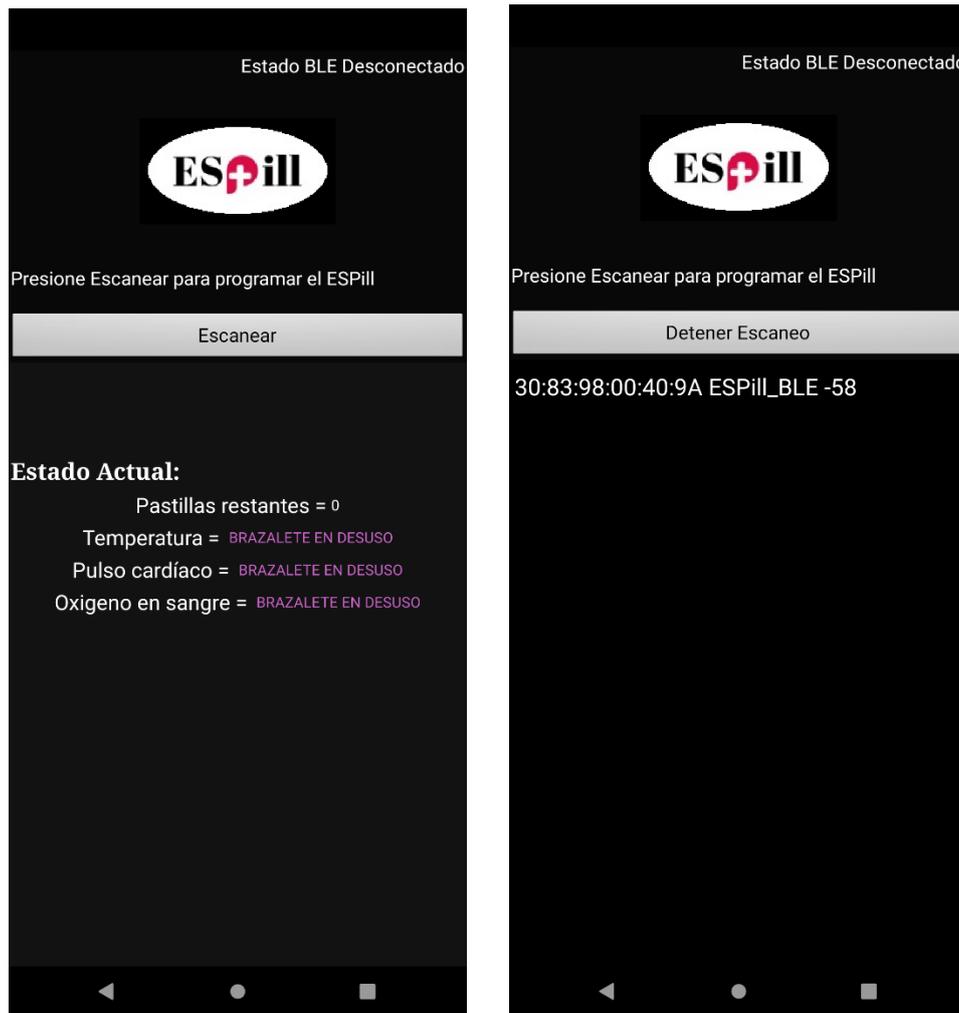


Figura 23. Pantalla principal y pantalla de escaneo – ESPill APP.

Una vez realizada la conexión, se habilitará la pantalla de programación y desconexión del pastillero. También existe la posibilidad de detener el escaneo y volver a la primera pantalla.

Para establecer la programación de los compartimentos se debe presionar el botón “programar” de la imagen 29 izquierda.

A continuación (imagen 29 derecha), se visualizan los distintos bloques de texto, donde se deben indicar el día (numérico de 1 a 31), hora (24 horas) y minutos. Opcionalmente es posible anotar una nota vinculada a cada compartimento.

Descripción de los botones de la pantalla de programación:

- Salir: descartar toda la programación realizada y volver a la pantalla de programación y desconexión.
- Atrás: retroceder un compartimento, descartando la programación del compartimento actual.

- Siguiente: ir al próximo compartimento, aceptando la programación del compartimento actual, teniendo la opción de volverlo a programar si se pulsa el botón “atrás”.
- Aceptar: es posible finalizar la programación parcial de compartimentos disponibles. Esto hará que el resto de los compartimentos no programados sean omitidos, volviendo a la pantalla de programación y desconexión.

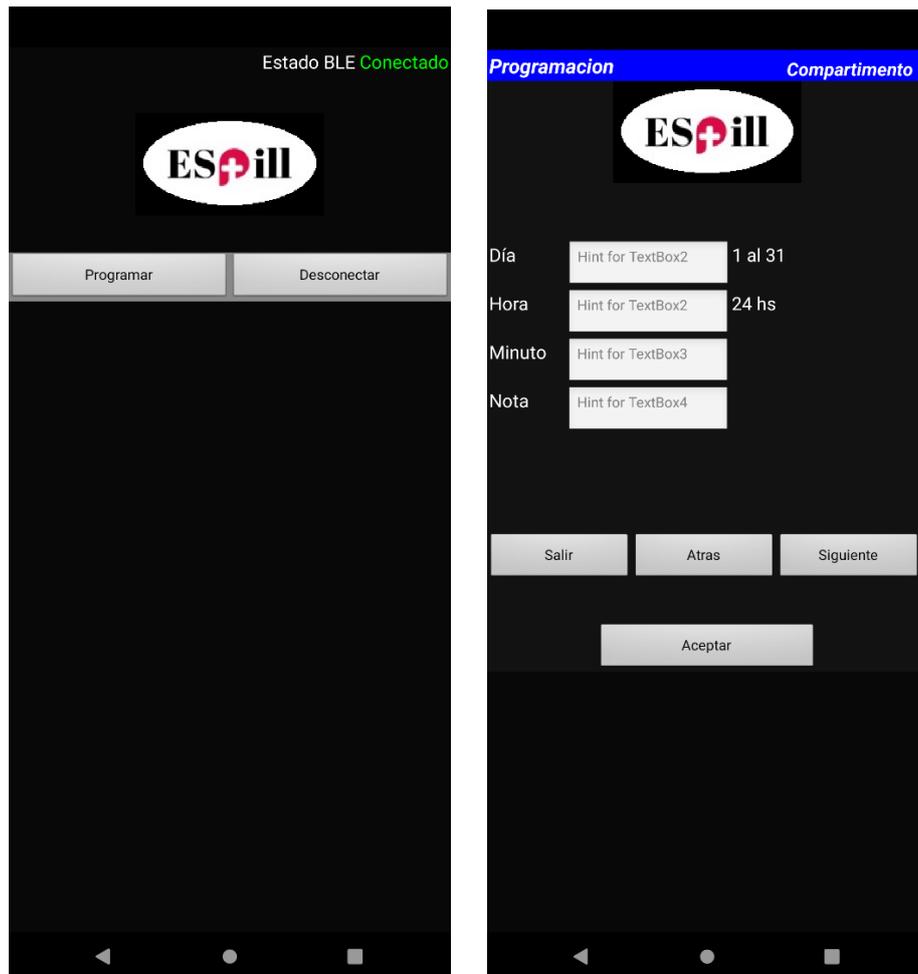


Figura 24. Pantalla de conexión y programación de compartimentos – ESPill APP.

Una vez finalizada la programación de las dosis, se debe desconectar la aplicación del pastillero, el cual quedará programado.

4.5. Servidor ThingSpeak

Es muy utilizado para aplicaciones IoT porque permite al usuario recopilar y visualizar las mediciones tomadas por el sistema. Esta plataforma proporciona un servidor en la nube

donde los usuarios pueden enviar datos desde sus dispositivos utilizando una variedad de protocolos de comunicación, como HTTP, MQTT y TCP/IP.

En este caso, se crean canales personalizados en ThingSpeak para organizar y almacenar las mediciones de los sensores. Los datos enviados a través de los canales son almacenados y visualizados en forma de gráficos y tablas. En ocasiones, es posible compartir públicamente los resultados. Para ingresar al canal del servidor se requiere saber las credenciales:

- Channel ID.
- API Key.

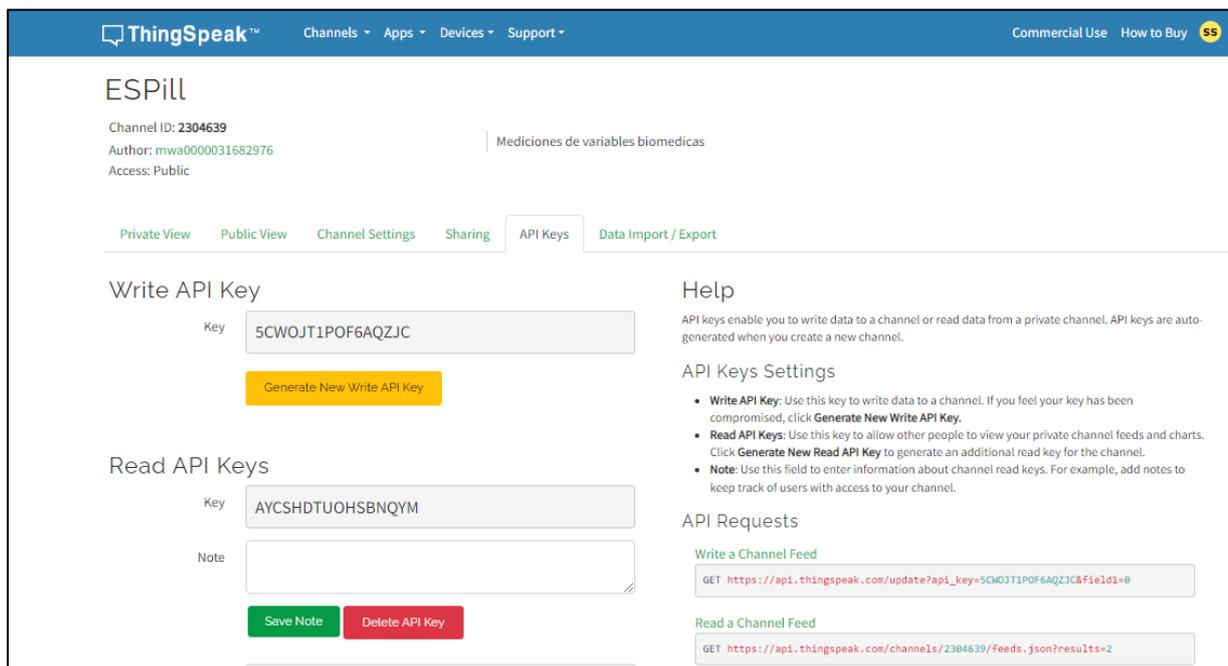


Figura 25. Credenciales para ThingSpeak - ESPill.

5.1. Diagramas de flujo de los Módulos

Un diagrama de flujo, también conocido como diagrama de actividades, es una herramienta gráfica que se utiliza para representar de manera visual un algoritmo o proceso. Consiste en una serie de pasos estructurados y conectados entre sí mediante flechas o líneas que indican la dirección del flujo del proceso. Cada paso se representa mediante figuras geométricas, las cuales se enlazan para formar un mapa visual del proceso que permite su revisión y comprensión.

5.1.1. Diagrama de flujo del Pastillero

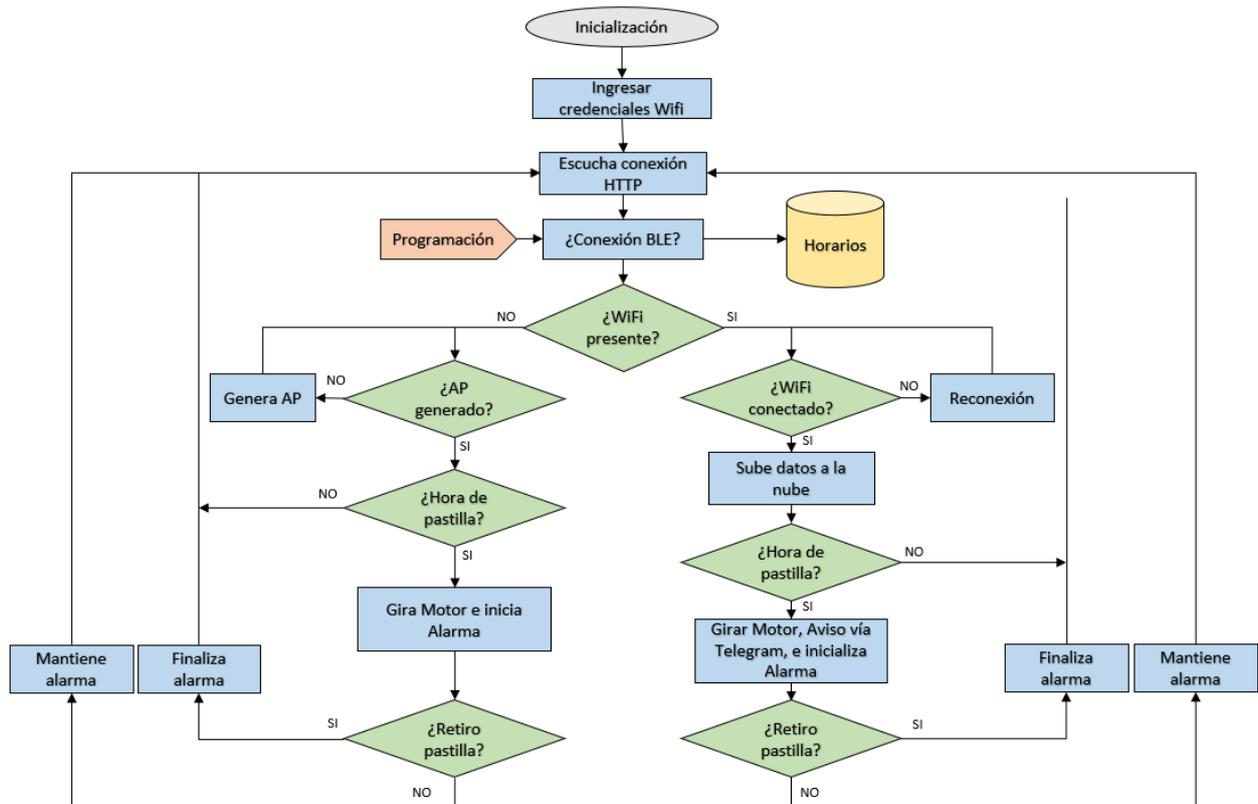


Figura 26. Diagrama de flujo del Pastillero.

5.1.2. Diagrama de flujo del Brazalete

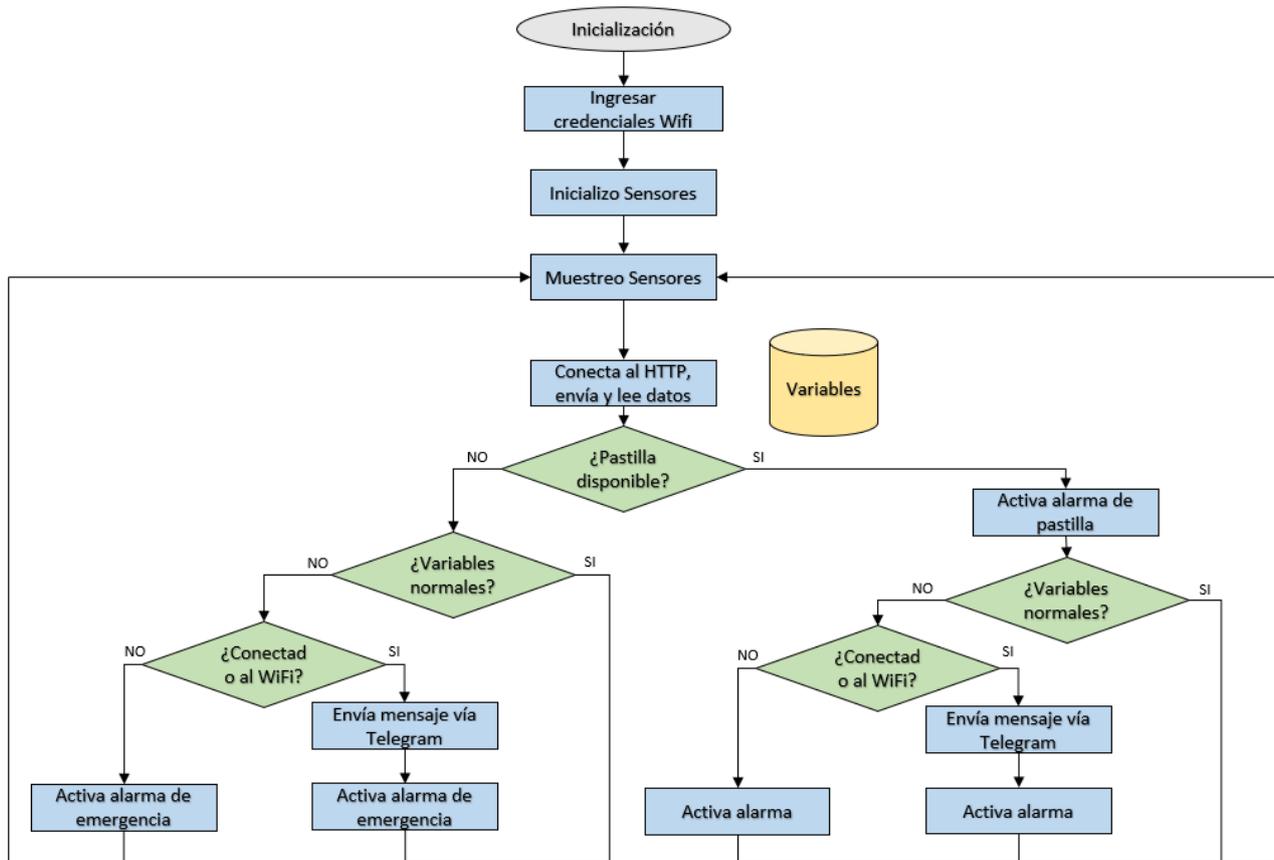


Figura 27. Diagrama de flujo del Brazalete.

5.1.3. Diagrama de flujo de App. Móvil

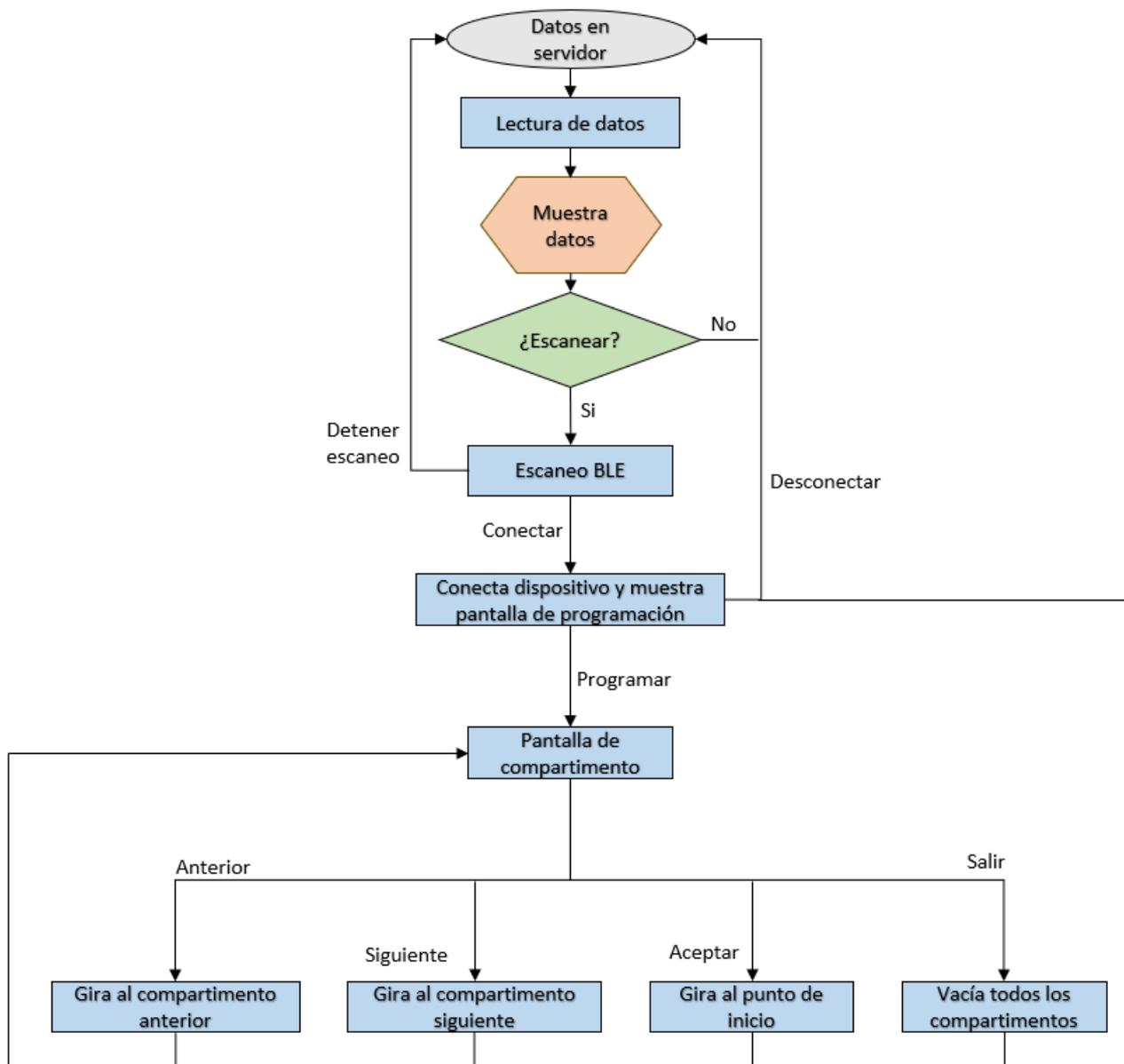


Figura 28. Diagrama de flujo de la App. Móvil.

6. Programación

6.1. Brazalete

Para la programación se utilizan los dos núcleos del microcontrolador. El núcleo 1 abarca tareas de las funciones *setup()* y *loop()*, mientras que el 0 hace el muestreo de los sensores de pulso cardíaco y temperatura. Se hace uso de los dos núcleos porque el MAX30100 necesita una frecuencia de muestreo tal que la cantidad de errores en la lectura sea mínima.

El bloque que ejecuta el núcleo 0 es:

```

47.  while (1) {
48.
49.    pox.update();
50.
51.
52.    if (millis() - tsLastReport > REPORTING_PERIOD_MS) {
53.
54.      if (xSemaphoreTake(xSensores, portMAX_DELAY) == pdTRUE) {
55.        Datos_auxiliar2.temperatura = bmp.readTemperature();
56.        Datos_auxiliar2.ox_sangre = pox.getSpO2();
57.        Datos_auxiliar2.pulsaciones = pox.getHeartRate();
58.      }
59.      xSemaphoreGive(xSensores);
60.
61.      tsLastReport = millis();
62.    }
63.  }

```

El muestreo periódico permite una actualización del sensor mediante la función `pox.update()`. Para utilizar la variable en memoria compartida, se hace uso de un semáforo, de manera tal que, si el núcleo 0 toma la variable para escribirla, el núcleo 1 no podrá leerla hasta que no se libere, y viceversa.

6.1.1. Función `setup()`

Se realizan las inicializaciones de los servicios a utilizar, tales como:

- MAX30100: obtención de los valores de pulso y oxigenación sanguínea.
- Wifi manager: conexión inicial al wifi.
- Bot de Telegram: notificaciones y alarmas.
- BMP280: obtener la temperatura corporal.
- Display OLED: dar conocimiento del estado actual.

6.1.2. Función loop()

Se evalúa si es posible tomar (si está disponible) el semáforo llamado `xSensores`, devolviendo un valor `pdTRUE` si se puede. Esto se hace para restringir el acceso del otro núcleo a recursos compartidos.

Cuando se accede a las variables, se genera una copia de los datos de los sensores. Luego con la instrucción `xSemaphoreGive()` se libera el semáforo, para que las demás tareas puedan acceder al recurso.

```

141. void loop() {
142.
143.   if (xSemaphoreTake(xSensores, portMAX_DELAY) == pdTRUE) {
144.     Serial.println("Nucleo -> " + String(xPortGetCoreID()) + " Actualizo variables");
145.     Datos_brazalete.temperatura = Datos_auxiliar2.temperatura;
146.     Datos_brazalete.ox_sangre = Datos_auxiliar2.ox_sangre;
147.     Datos_brazalete.pulsaciones = Datos_auxiliar2.pulsaciones;
148.   }
149.   xSemaphoreGive(xSensores);

```

Tras copiar los datos de los sensores en una variable propia del núcleo 1, se realiza una lógica para verificar si el valor obtenido es coherente.

Si se recibe un valor nulo, significa que no se esté utilizando el brazalete, o que haya habido un error en el censado, por lo que se descarta, aumentando un contador filtro.

```

151.   if (filtro.temperatura > 0 && filtro.pulsaciones > 0 && filtro.ox_sangre > 0) {
152.
153.     muestras[i].temperatura = filtro.temperatura;
154.     muestras[i].pulsaciones = filtro.pulsaciones;
155.     muestras[i].ox_sangre = filtro.ox_sangre;
156.
157.     Serial.println("Nucleo -> " + String(xPortGetCoreID()) + " i: " + i + " Temp: " +
158.     muestras[i].temperatura + " Puls: " + muestras[i].pulsaciones + " Ox: " +
159.     muestras[i].ox_sangre);
160.     i++;
161.   } else {
162.     contador_filtro++;
163.   }

```

Si el contador filtro llega a un valor determinado, la muestra se considera coherente y es tomada en cuenta. Luego, con 10 muestras tomadas, se realiza el promedio. Los valores promedio son mostrados en el display y enviados mediante HTTP al pastillero.

A continuación, se evalúa el estado de la conexión a Wifi:

```

204. if (wifiMulti.run() != WL_CONNECTED) {
205.   Serial.println("Wifi y AP desconectado");
206.   flag_pox_init = 0;
207.   delay(300);
208. } else {
209.
210.   if (WiFi.SSID() == ssidString) {

```

```
211.
212.     wifi_conectado = 1;
213.     if (flag_pox_init == 0) {
214.         pox.begin();
215.         flag_pox_init = 1;
216.     }
217.     Serial.println("Conectado al WiFi");
218.
219.     if (mensaje_ppal_telegram == 0) {
220.         miBot.sendMessage(-998425360, "Brazalete conectado a WiFi");
221.         mensaje_ppal_telegram = 1;
222.     }
223. } else {
224.
225.     wifi_conectado = 0;
226.     mensaje_ppal_telegram = 0;
227.     Serial.println("Conectado al AP");
228.
229.     if (flag_pox_init == 0) {
230.         pox.begin();
231.         flag_pox_init = 1;
232.     }
233. }
234. }
```

En este punto se desprenden 3 posibilidades:

- Sistema conectado al Wifi deseado, gestionado por la librería *WifiManager()*.
- Ausencia de Wifi, por lo que el brazalete se conectó al AP que le genera el pastillero.
- Ausencia de ambas redes.

Si la conexión al Wifi fue exitosa, se activan las alertas de Telegram, pero si está conectado al AP del pastillero no será posible enviar alertas, quedando como única comunicación, la que tiene con el pastillero. En ausencia de ambas, sólo quedarán activas las alertas de vibración y de display en el brazalete.

Este bloque testea constantemente la conexión, y se conecta a la que se encuentre disponible; es tarea del pastillero escanear y verificar las redes disponibles.

Luego, se ejecuta la instrucción de actualización, la cual envía los valores muestreados al display, al pastillero y verifica la alarma. Esta última está programada para activarse cuando suceden 5 valores anómalos seguidos, evitando así las falsas alertas. El tiempo de envío de alertas por Telegram ante una emergencia es prácticamente instantáneo.

6.2. Pastillero

Utiliza ambos núcleos para que la alarma no dependa del *loop()* principal del programa. De similar manera, se podría realizar utilizando FreeRTOS. El código del núcleo 0 solo obtiene el estado de la variable compartida “alarma”, a través del uso de un semáforo.

```
47.     if (xSemaphoreTake(xAlarma, portMAX_DELAY) == pdTRUE) {
48.         alarma = alarma_act;
49.     }
50.     xSemaphoreGive(xAlarma);
```

6.2.1. Función setup()

Se hacen las inicializaciones de los servicios a utilizar:

- Bluetooth BLE: programación de la aplicación móvil.
- Sensor de proximidad: detección del usuario al retirar las pastillas.
- Wifi mánager: conexión inicial al wifi.
- Bot de Telegram: notificaciones y alarmas.
- Motor paso a paso: movimiento del pastillero.
- Servidor HTTP: comunicación con el brazaletes.
- ThingSpeak: subir los datos a la nube y posteriormente verlos en la aplicación.
- mDNS: resolver el dominio local a la IP del pastillero sin necesidad de un servidor DNS.
- RTC: obtención de la hora actual desde un servidor NTP.

Una vez se inicializa el Wifi Manager, y se obtienen las credenciales se guardarán en dos variables para manejar el cambio de conexión en ausencia de Wifi.

```
135.     ssidString = wm.getWiFiSSID();
136.     passwordString = wm.getWiFiPass();
```

Así, se logra mayor libertad a la hora de verificar las conexiones, pasando a modo estación y conectando a la red con la instrucción de la librería CTbot de Telegram.

```
145.     WiFi.disconnect();
146.     WiFi.mode(WIFI_MODE_STA);
147.     miBot.wifiConnect(ssidString, passwordString);
148.     delay(500);
149.     miBot.setTelegramToken(token_bot);
```

Una vez establecida la conexión a internet, se inicializa el servidor HTTP para comunicarse con el brazaletes, y se agrega el servicio a mDNS (multicast Domain Name System), para que la IP dinámica que obtenga el pastillero sea transparente, y pueda utilizarse un dominio.

```
168.     server.begin();
169.     Serial.println("TCP server started");
170.     MDNS.addService("http", "tcp", 80);
```

6.2.2. Función loop()

Manejo de la conexión Wifi:

```

202.   if (WiFi.status() == WL_CONNECTED) {
203.       success = Ping.ping("149.154.167.220", 3);
204.       if (!success) {
205.           conexion_internet = 0;
206.       } else {
207.           conexion_internet = 1;
208.           configTime(gmtOffset_sec, daylightOffset_sec, ntpServer);
209.           struct tm timeinfo;
210.           if (getLocalTime(&timeinfo)) {
211.               rtc.setTimeStruct(timeinfo);
212.           }
213.       }
214.   } else {
215.       conexion_internet = 0;
216.       if (currentMillis - previousScanMillis >= scanInterval || auxiliar2 == 0) {
217.           auxiliar2 = 1;
218.           uint8_t o = 0, numNetworks;
219.           numScannedNetworks = 0;
220.           numNetworks = WiFi.scanNetworks();
221.
222.           for (int t = 0; t < numNetworks && numScannedNetworks < MAX_NETWORKS; t++)
223.           {
224.               scannedNetworks[numScannedNetworks] = WiFi.SSID(t);
225.               numScannedNetworks++;
226.           }
227.
228.           for (int t = 0; t < numScannedNetworks; t++) {
229.               Serial.printf("%d: %s, RSSI: %d dBm\n", t + 1, WiFi.SSID(t).c_str(),
WiFi.RSSI(t));
230.               if (scannedNetworks[t] == ssidString) {
231.                   flag_AP = 0;
232.                   Serial.println(ssidString);
233.                   WiFi.mode(WIFI_MODE_STA);
234.                   miBot.wifiConnect(ssidString, passwordString);
235.                   delay(500);
236.                   miBot.setTelegramToken(token_bot);
237.                   miBot.sendMessage(-998425360, "OK");
238.                   return;
239.               } else {
240.                   Serial.print("No se encontró la red Wifi");
241.                   flag_AP = 1;
242.               }
243.           }
244.
245.           if (flag_AP && WiFi.status() != WL_CONNECTED) {
246.               WiFi.mode(WIFI_MODE_AP);
247.               WiFi.softAP("ESP32", "password");
248.               delay(500);
249.           }
250.           previousScanMillis = currentMillis;
251.       }
252.   }
253.

```

Si el sistema está conectado a la red Wifi, se hace ping a la IP pública de los servidores de Telegram para determinar si efectivamente hay conexión a Internet. Si este es el caso, se

activan todas las instrucciones que dependen de Internet (NTP, ThingSpeak y Telegram). Esta verificación se realiza porque es posible estar conectado a Wifi sin tener acceso a Internet, y cuando se utiliza Telegram o algún otro servicio que deba tener disponibilidad externa, se tendrá un tiempo muerto importante en el algoritmo. Una vez asegurado que hay conexión a Internet, se actualiza la hora con el servidor NTP.

Si no hay conexión a Wifi, el pastillero se configura en modo AP, de manera que el brazalete pueda conectarse a él y no se detenga la transferencia de datos. Cada cierto tiempo, el pastillero cambia a modo estación y escanea las redes Wifi para intentar la reconexión a Wifi. Si encuentra la red ingresada, se conecta y continua la comunicación del sistema.

Manejo de la programación de los compartimentos:

Una vez establecida la conexión Bluetooth BLE en el pastillero, el sistema espera los valores numéricos de los compartimentos, siendo responsabilidad de la aplicación manejar el buffer de datos programados asociados al equipo. Luego de confirmar la programación de los compartimentos, se transfieren los datos recibidos desde el buffer de programación auxiliar a otro designado para preservar los datos finales de programación.

En el código se contempla un tiempo de seguridad para el caso que el usuario no realice cambios mientras la aplicación está emparejada. Pasado este tiempo, se desconecta y continúa con el bucle principal.

Manejo de la comunicación del sistema:

Para la comunicación se utiliza HTTP, por la simplicidad en el uso y la adecuada adaptación al tipo de transferencia de datos que deseamos realizar en este proyecto. A continuación, se encuentra el bloque de código encargado del procesamiento de datos vía HTTP:

```

377.  client_Wifi = server.available();
378.
379.  if (client_Wifi) {
380.
381.      while (client_Wifi.connected() && !client_Wifi.available()) {
382.          delay(1);
383.      }
384.      Serial.println(".");
385.
386.      String req = client_Wifi.readStringUntil('\r');
387.      Serial.println("Solicitud recibida: " + req);
388.
389.      Datos_pastillero.pastillas = pastillas;
390.      Datos_pastillero.alarma = 0;
391.
392.      client_Wifi.print(String(Datos_pastillero.pastillas));
393.      client_Wifi.print(',');
394.      client_Wifi.println(String(Datos_pastillero.alarma));
395.
396.      char *temp;
397.      char *temperature = strtok_r(const_cast<char *>(req.c_str()), ",", &temp);
398.      char *pulsations = strtok_r(nullptr, ",", &temp);
399.      char *ox_sangre = strtok_r(nullptr, ",", &temp);
400.
401.      if (temperature != nullptr && pulsations != nullptr && ox_sangre != nullptr) {

```

```

402.     Datos_brazalete.temperatura = atof(temperature);
403.     Datos_brazalete.pulsaciones = atof(pulsations);
404.     Datos_brazalete.ox_sangre = atof(ox_sangre);
405.     } else {
406.     Datos_brazalete.temperatura = 0;
407.     Datos_brazalete.pulsaciones = 0;
408.     Datos_brazalete.ox_sangre = 0;
409.     }
410.
411.     client_Wifi.stop();
412.
413.     }
414.

```

El pastillero funciona como servidor y el brazaletes como cliente, por lo que el brazaletes es el encargado de iniciar la comunicación. El servidor esperará a recibir los datos. Si el dato está corrupto o tiene errores, simplemente se descarta.

El manejo de las alarmas consiste en un comparador que utiliza el tiempo actualizado proveniente del servidor NTP y lo contrasta con la información almacenada en el buffer de programación. Si es la hora programada para tomar pastillas de un compartimento específico, el sistema activa la alarma correspondiente y establece el indicador de alarma en 1. De esta manera, el núcleo 0 activará la alarma sonora y visual hasta que el sensor de movimiento detecte que el usuario haya retirado el medicamento.

```

439.     if (rtc.getTime("%d").toInt() == prog_datos[comp_actual].dia) {
440.         if (rtc.getTime("%H").toInt() == prog_datos[comp_actual].hora) {
441.             if (rtc.getTime("%M").toInt() == prog_datos[comp_actual].minuto && alarma == 0) {
442.                 Serial.println("Nucleo -> " + String(xPortGetCoreID()) + " Actualizo a alarma");
443.                 if (xSemaphoreTake(xAlarma, portMAX_DELAY) == pdTRUE) {
444.                     alarma = 1;
445.                 }
446.                 xSemaphoreGive(xAlarma);
447.                 myStepper.step(128);
448.                 anular_retencion();
449.                 if (conexion_internet == 1) {
450.                     mensaje = "Tomar pastilla: " + String(prog_datos[comp_actual].nota);
451.                     miBot.sendMessage(-998425360, mensaje);
452.                 }
453.                 comp_actual++;
454.                 pastillas--;
455.             }
456.         }
457.     }

```

Cuando el motor finaliza el giro y queda en su posición final, sus bobinas se energizan creando una fuerza de retención. Esto provoca un consumo de energía innecesario, producto del calentamiento del sistema. Por esta razón, con la función *anular_retención()*, se reduce el consumo del equipo.

```

458. void anular_retencion(){
459.     delay(200);
460.     digitalWrite(IN1, LOW);
461.     digitalWrite(IN2, LOW);
462.     digitalWrite(IN3, LOW);
463.     digitalWrite(IN4, LOW);
464. }

```

6.3 Aplicación

Previo a la conexión Bluetooth entre el dispositivo móvil y el pastillero, la programación de la aplicación comienza indicando las variables que se extraen de la nube (servidor ThingSpeak).

```

when Web1 .GetText
do
  uri responseCode responseType responseContent
  if get responseCode == 200
  then
    set global json to look up in pairs key "feeds"
    pairs call Web1 .JsonTextDecode jsonText get responseContent
    notFound "No encontrado"
    set global Pastillas to select list item list select list item list select list item list get global json
    index 2 index 3 index 1
    set global Temperatura to select list item list select list item list select list item list get global json
    index 2 index 4 index 1
    set global Latidos to select list item list select list item list select list item list get global json
    index 2 index 5 index 1
    set global Oxigeno to select list item list select list item list select list item list get global json
    index 2 index 6 index 1
  
```

Figura 29. Programación en bloques – petición web.

Esta función se encarga de procesar el JSON obtenido de la página web de ThingSpeak, separando los valores y almacenándolos en variables para su posterior visualización. En el bloque de la Figura 29 se observa el proceso de extracción de los valores.

Posteriormente, si existe una conexión Bluetooth, se muestran los botones de programación y desconexión para interactuar con el usuario. El bloque se puede ver a continuación:

```

when B_Programar .Click
do
  set global pro to 0
  set Principal .Visible to false
  set Prog1 .Visible to true
  set Label_No_compartimento .Text to join "Compartimento"
  get global pro + 1
  if BLE .IsDeviceConnected
  then
    call BLE .WriteStringsWithResponse
    serviceUuid get global UUID_Service
    characteristicUuid get global UUID_Char1
    utf 16 false
    values join "/prog"
    " "
  
```

Figura 30. Programación en bloques – Botón programar.

Los botones transmiten una instrucción determinada al microcontrolador del pastillero. Para iniciar el modo de programación se envía la cadena “/prog.”.

Luego, los botones “aceptar”, “atrás”, “siguiente” y “salir” envían las instrucciones para que el motor gire y se almacenen en el buffer de programación las variables ingresadas por el operador. Los diagramas en bloque correspondientes son los de las Figuras 31, 32, 33 y 34:

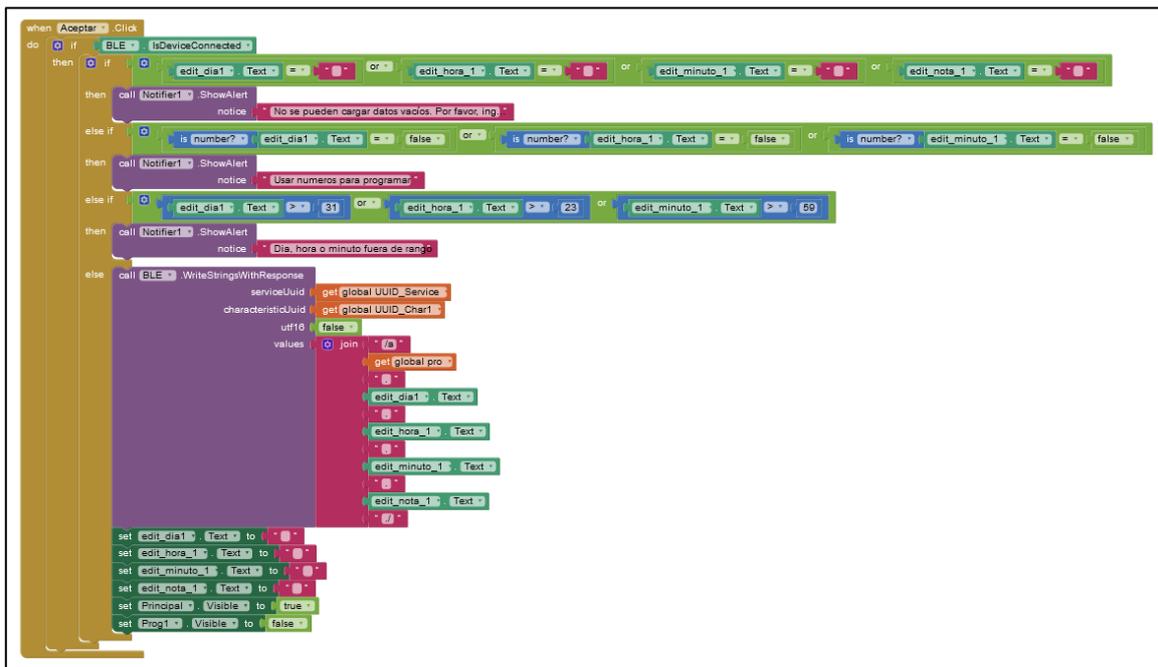


Figura 31. Programación en bloques – botón aceptar.

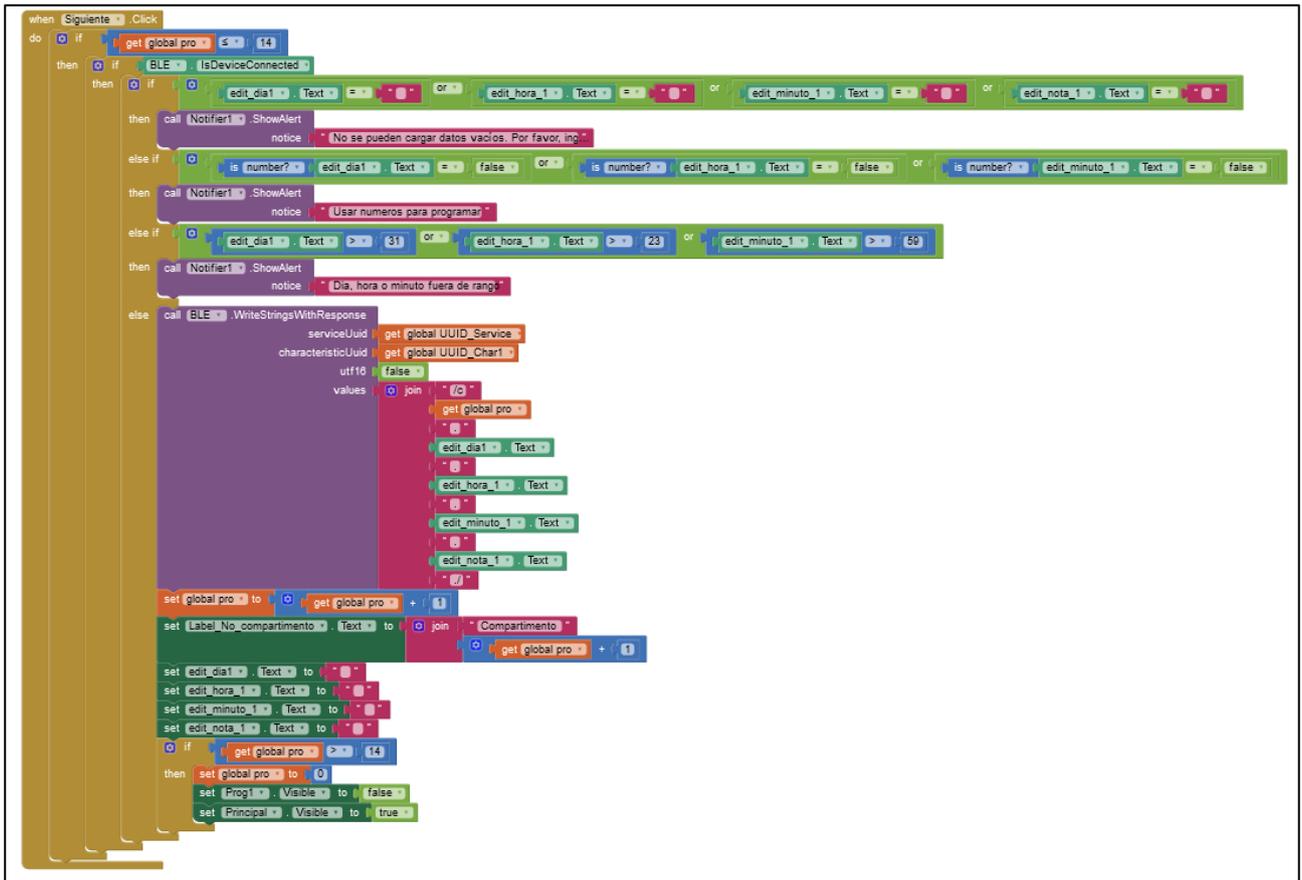


Figura 32. Programación en bloques – botón siguiente.

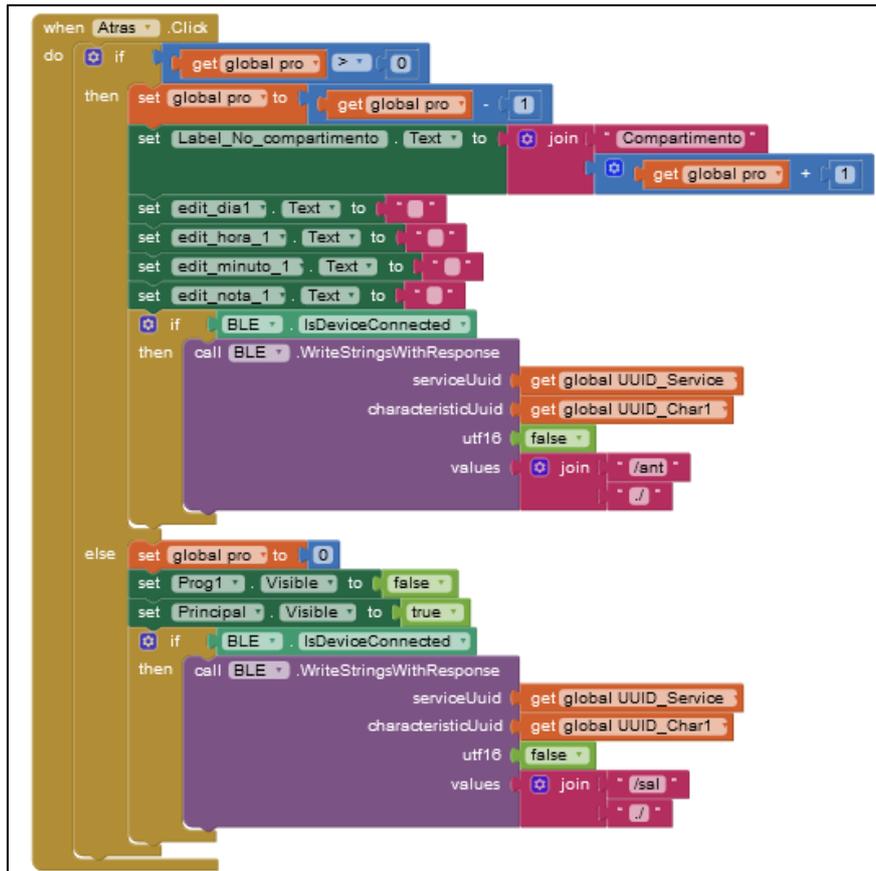


Figura 33. Programación en bloques – botón atrás.

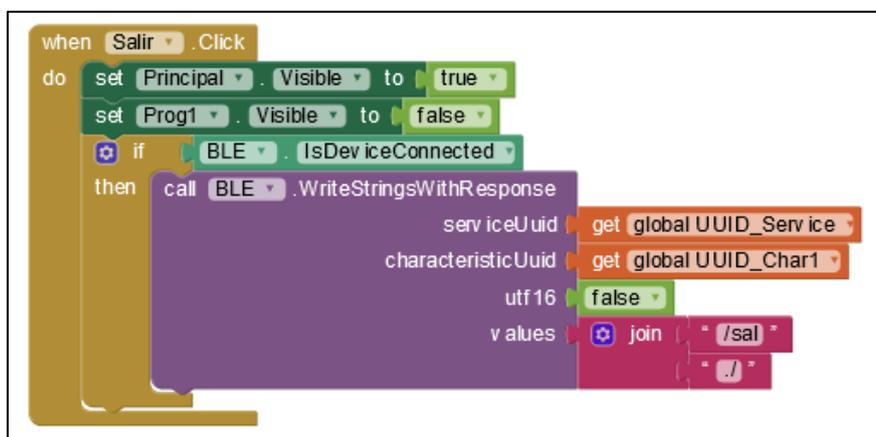


Figura 34. Programación en bloques – botón salir.

Previo al envío del horario de programación, se realiza una verificación de coherencia del valor. Tanto la fecha como la hora se validan para asegurar que estén dentro del rango apropiado y sean valores numéricos válidos.

Es importante que el usuario sepa que la programación de los compartimentos debe seguir un orden específico porque el pastillero gira en una sola dirección. Por ejemplo, si la programación del compartimento 1 tiene una fecha establecida para el día 15, no será posible programar el compartimento 2 con una fecha anterior a ésta.

7. Pruebas finales

7.1. Mediciones ESPill

Una vez finalizada la etapa de diseño del hardware y la programación del sistema, se tomaron las mediciones con la persona sentada.

Tabla 5. Mediciones tomadas con el brazalete colocado en el brazo del usuario.

ESPill Temperatura corporal [° C]	ESPill Pulsaciones [ppm]	ESPill Ox. Sangre [%]
36,11	64,74	95,5
36,31	62,25	99,5
36,31	61,95	99,5
36,4	68,42	99,2
36,47	57,76	99
36,65	61,27	98,7
36,58	57,33	97,9
36,58	59,33	97,9
36,6	60,2	100
36,6	62,8	97,6
36,6	62,8	97,6
36,59	59,34	97,6
36,57	61,57	98,9
36,55	60,43	98,4
36,46	58,96	100
36,45	59,34	100
36,43	57,66	99,3
36,43	61,15	99,3
36,38	58,97	98,8
36,37	60	99,3
36,32	62,48	99

7.2 Contraste de las mediciones

Para comparar los resultados obtenidos con el brazalete, se hace uso de dos instrumentos de medición comerciales. Para el caso del pulso cardíaco y saturación de oxígeno en sangre, se recurre el equipo Contec CMS50DL, mientras que para comparar la temperatura corporal se emplea un termómetro digital, Franklin HZ6.

Comparación de temperatura corporal:



Figura 35. Equipos comerciales usados para contrastar las mediciones.

Para calcular el error relativo en todas las mediciones se utiliza la fórmula:

$$\%ERROR = \frac{|Valor_{MEDIDO} - Valor_{PATRÓN}|}{Valor_{PATRÓN}} * 100\%$$

En la tabla que sigue se ve el contraste de las mediciones de temperatura corporal entre el sistema ESPill y el termómetro comercial:

Tabla 6. Error relativo en la temperatura corporal.

ESPill Temperatura corporal [° C]	Termómetro digital HZ-6 [° C]	Error Relativo Porcentual [%]
36,11	35,8	0,87
36,31	35,9	1,14
36,31	36,2	0,30
36,4	36,2	0,55
36,47	36,3	0,47
36,65	36,3	0,96
36,58	36,2	1,05
36,58	36,1	1,33
36,6	36,2	1,10
36,6	36,4	0,55
36,6	36,3	0,83
36,59	36,4	0,52
36,57	36,5	0,19
36,55	36,5	0,14
36,46	36,4	0,16
36,45	36,3	0,41
36,43	36,2	0,64
36,43	36,3	0,36
36,38	36,2	0,50
36,37	36,3	0,19
36,32	36,4	0,22

En la Figura 36 se muestra la disparidad entre las mediciones de temperatura corporal de los equipos. El eje “x” representa la cantidad de mediciones realizadas, mientras que el eje “y” muestra los valores medidos en grados centígrados. La curva azul representa los datos obtenidos con el ESPill y la curva naranja corresponde a las temperaturas del instrumento tomado como referencia.

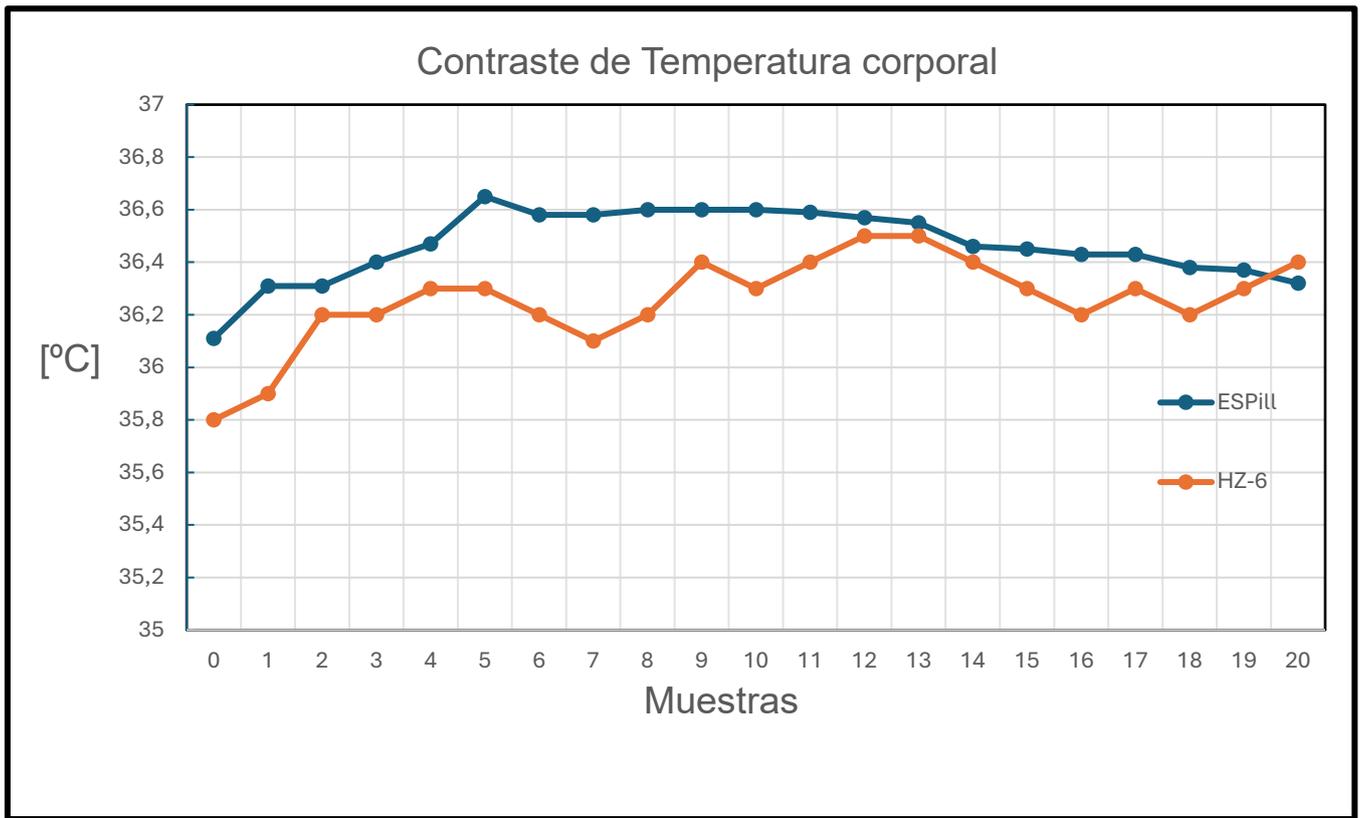


Figura 36. Gráfico comparativo de temperatura corporal.

Comparación de frecuencia cardíaca:

En la Tabla 7 se puede ver el contraste entre las mediciones de frecuencia cardíaca y el error relativo.

Tabla 7. Error relativo para las pulsaciones cardíacas.

ESPill Pulsaciones [ppm]	Oxímetro CMS50DL Pulsaciones [ppm]	Error Relativo Porcentual [%]
64,74	64	1,16
62,25	61	2,05
61,95	63	1,67
68,42	66	3,67
57,76	59	2,10
61,27	63	2,75
57,33	60	4,45
59,33	62	4,31
60,2	62	2,90
62,8	63	0,32
62,8	61	2,95
59,34	62	4,29
61,57	64	3,80
60,43	63	4,08
58,96	61	3,34
59,34	62	4,29
57,66	61	5,48
61,15	64	4,45
58,97	63	6,40
60	62	3,23
62,48	61	2,43

El gráfico comparativo que se desprende de comparar las pulsaciones cardíacas del brazalete con las del oxímetro comercial es el de la Figura 37:

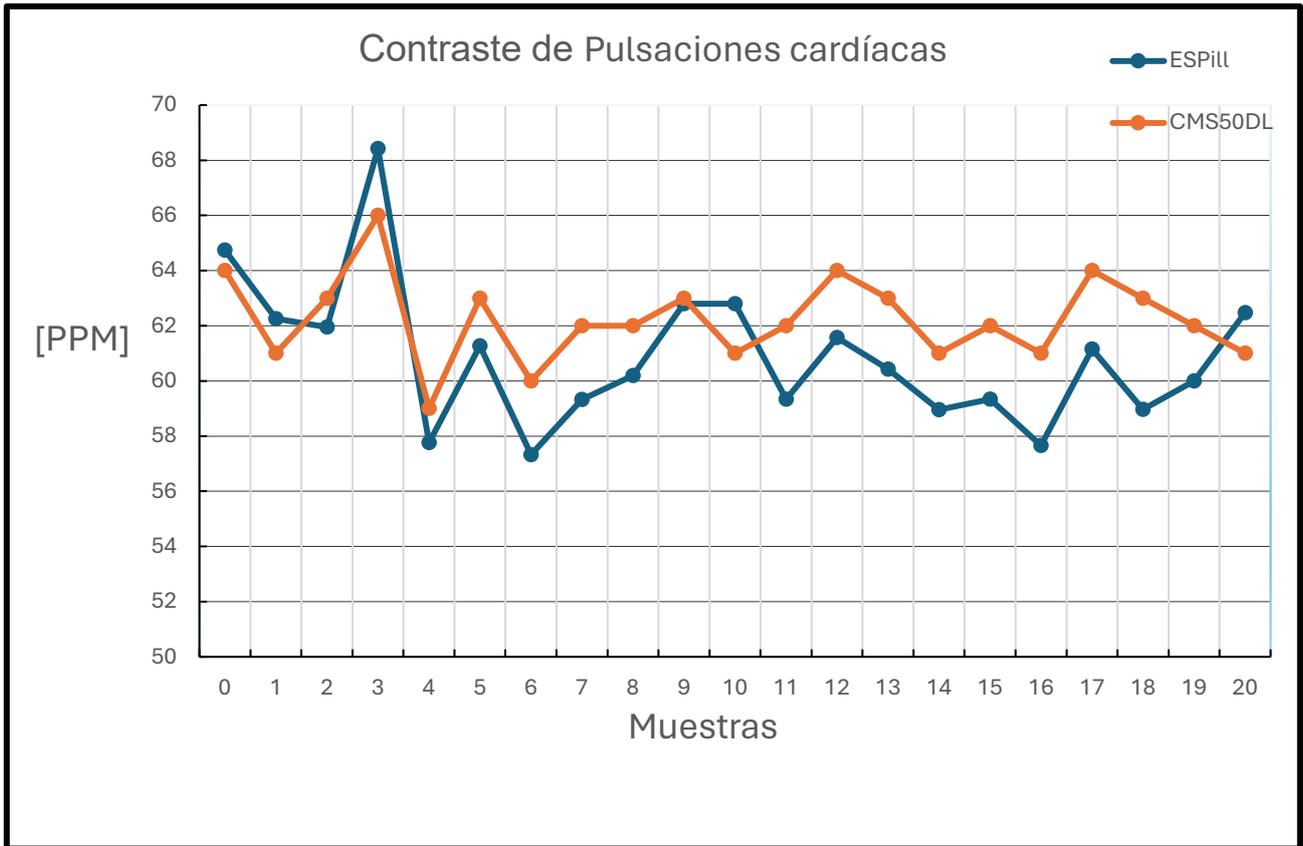


Figura 37. Gráfico comparativo de pulsaciones cardíacas.

Comparación de oxigenación en sangre:

En la Tabla 8 se puede ver el contraste para las mediciones de oxigenación de la sangre y el error relativo.

Tabla 8. Error relativo para el oxígeno en sangre.

ESPill Ox. Sangre [%]	Oxímetro CMS50DL Ox. Sangre [%]	Error Relativo Porcentual [%]
95,5	98	2,55
99,5	98	1,53
99,5	97	2,58
99,2	98	1,22
99	98	1,02
98,7	98	0,71
97,9	97	0,93
97,9	98	0,10
100	98	2,04
97,6	96	1,67
97,6	98	0,41
97,6	98	0,41
98,9	97	1,96
98,4	96	2,50
100	97	3,09
100	97	3,09
99,3	97	2,37
99,3	98	1,33
98,8	98	0,82
99,3	98	1,33
99	98	1,02

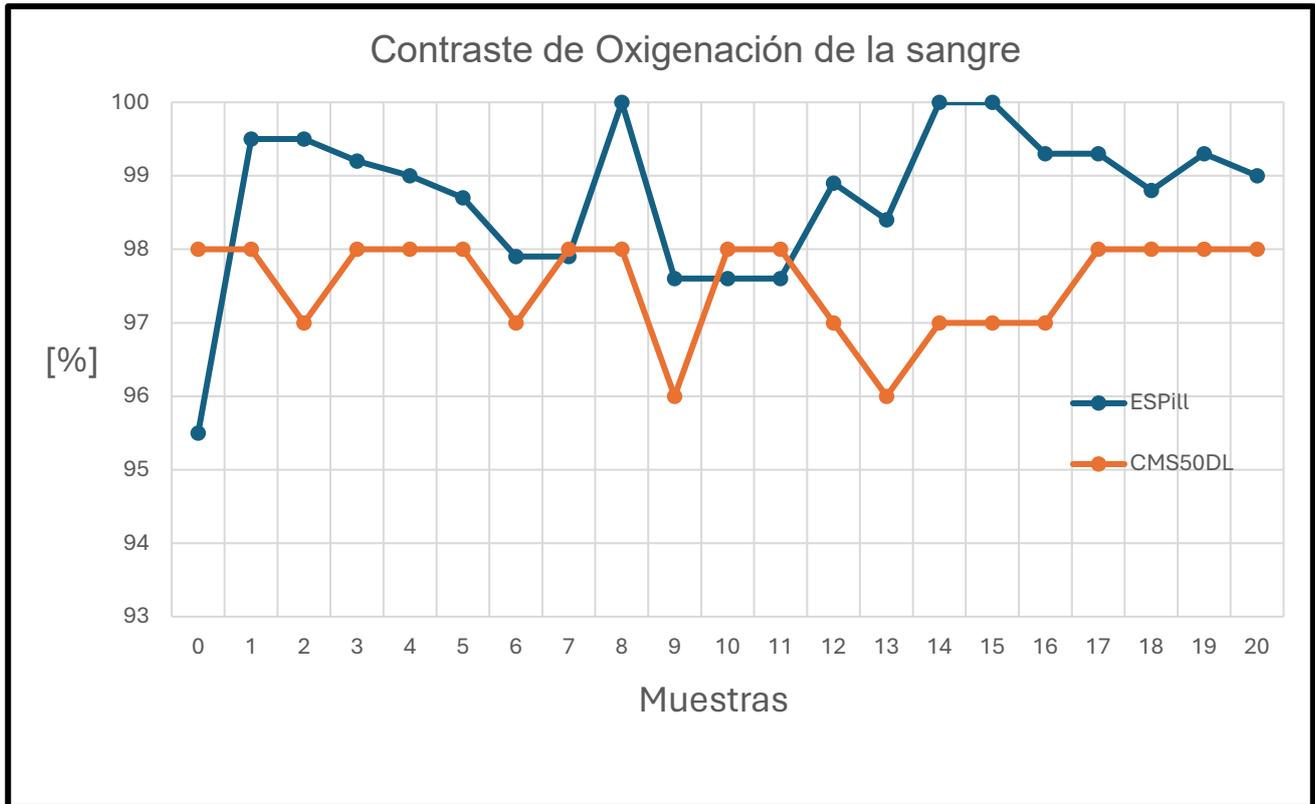


Figura 38. Gráfico comparativo del oxígeno en sangre.

Viendo las comparaciones se observa que el mayor error relativo se obtuvo en las mediciones de frecuencia cardíaca, siendo de 6,4%. Pero en términos de valores de las PPM un error relativo de esa magnitud es aceptable porque representa una diferencia de solamente 5 ppm, aproximadamente, respecto del valor medido con el oxímetro comercial. En cambio, si se hubiera obtenido el mismo error en las mediciones de temperatura corporal y de oxigenación de la sangre, no resultan aceptables porque, en términos de unidades, representaría grandes diferencias respecto del valor de referencia.

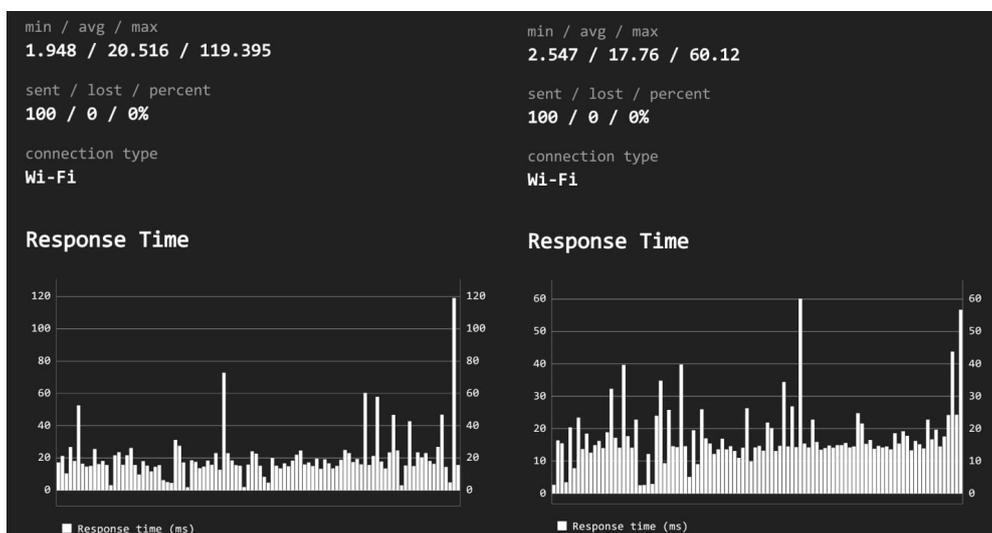
Si bien los equipos comerciales utilizados para contrastar las mediciones son no invasivos como el ESPill, se deduce que las diferencias en las mediciones se deben en gran parte al lugar en donde se midieron. Es decir, para medir los valores del equipo desarrollado se hace desde el lóbulo del oído; mientras que el termómetro digital empleado se coloca en la zona axilar. Lo mismo sucede con el oxímetro comercial, ya que se coloca en la punta del dedo, pero las mediciones del ESPill se tomaron en el lóbulo de la oreja, a través del broche del brazalete. Si bien los resultados son aceptables se pueden mejorar midiendo y contrastando los signos vitales en la misma zona del cuerpo.

7.3 Cobertura del Modo AP

Las pruebas de conexión entre el brazalete y el pastillero pretenden determinar la distancia máxima a la que ambos dispositivos pueden permanecer comunicados.

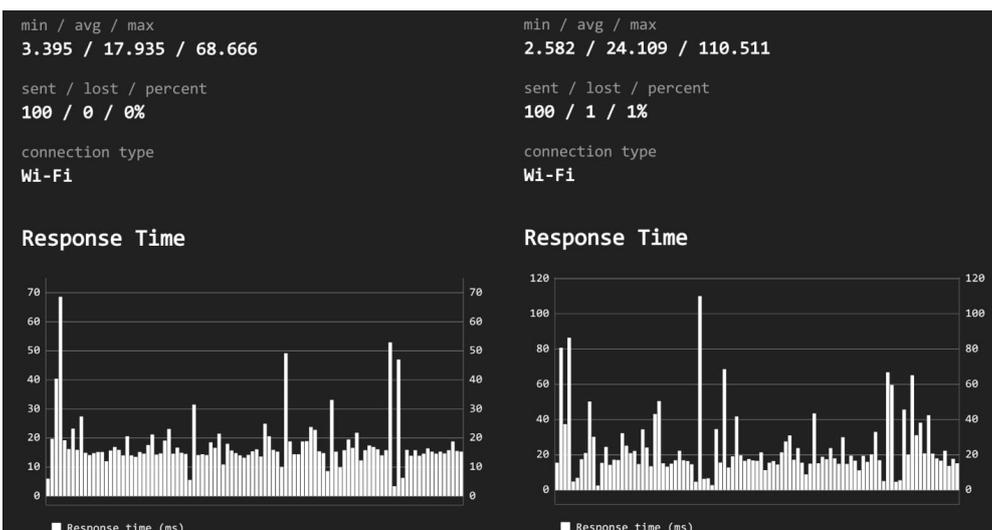
En estas pruebas se hace uso del comando “ping” para enviar paquetes de tamaño fijo con el fin de verificar la conexión con los hosts de la red. Durante estos ensayos, el pastillero permanece en modo AP, ya que si se hicieran en modo estación se estaría evaluando la red Wifi. Por lo que las distancias alcanzadas dependen principalmente del hardware del router utilizado.

Imágenes de las mediciones más representativas:



5 metros sin barreras

7 metros sin barreras



8 metros sin barreras

7 metros 1 barrera

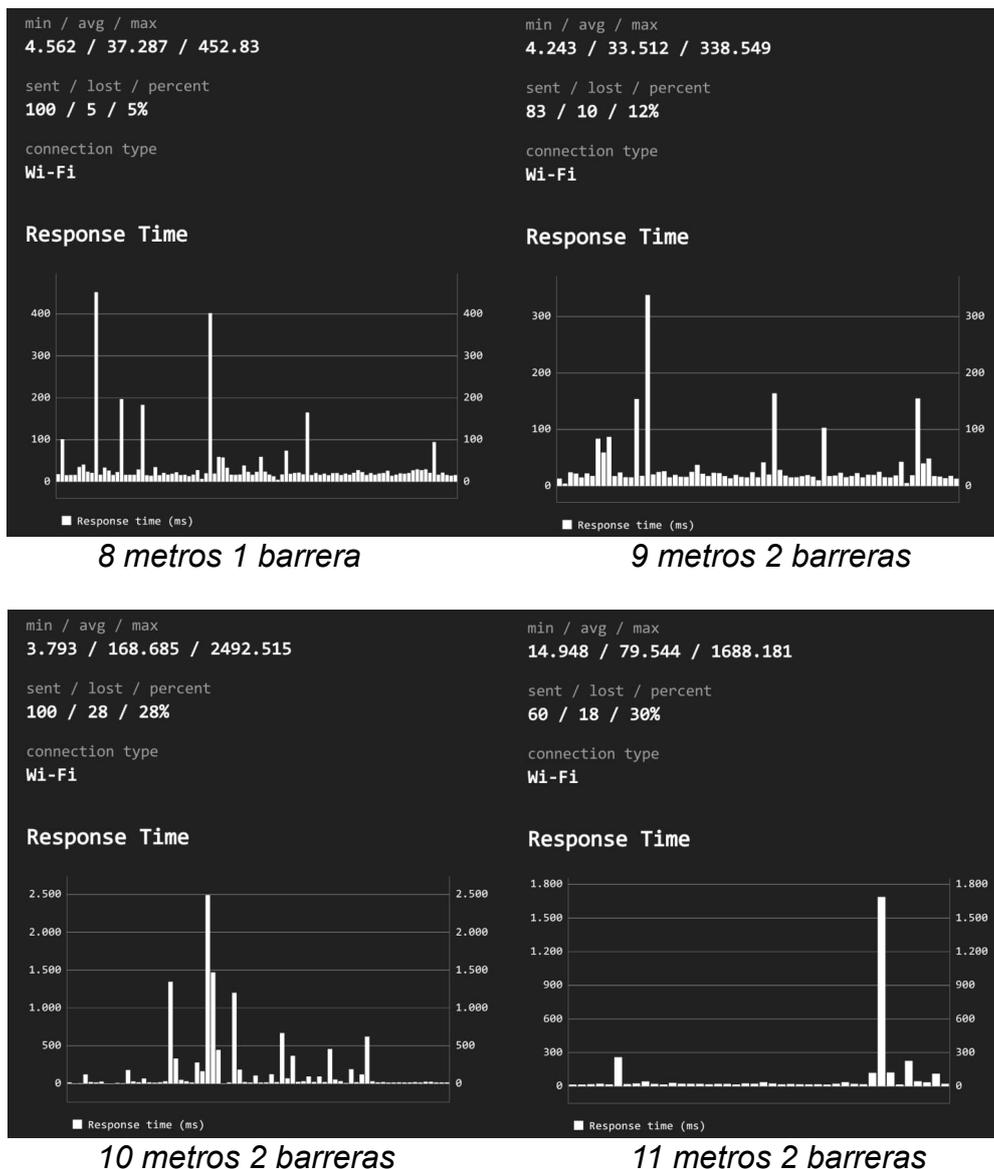


Figura 39. Imágenes de testeo de RED representativas

De la Figura 39 se observa que se enviaron 60 paquetes antes de que la conexión se interrumpiera debido a la pérdida de cobertura del AP. En comparación con distancias más cortas, se nota una disminución en la cantidad de paquetes enviados y una pérdida mayor. A distancias mayores, la calidad de la conexión se ve afectada, debido a obstáculos físicos o interferencias electromagnéticas presentes.

En la Tabla 9 se ven los resultados obtenidos.

Tabla 9. Testeo de la cobertura de la red AP.

Distancia [m]	5	7	8	7	8	9	10	11
Barreras	0	0	0	1	1	2	2	2
Enviados [%]	100	100	100	100	100	100	100	60
Recibidos [%]	100	100	100	99	95	90	72	82
Tiempo promedio de transmisión [mseg]	20	17	18	24	37	33	168	79
Pérdidas [%]	0	0	0	1	5	12	28	30

En resumen, la transmisión de datos sin pérdidas importantes no puede garantizarse más allá de los 8 o 9 metros, dependiendo de la presencia y densidad de obstáculos presentes en el entorno.

8. Conclusiones

El proyecto ESPill ha sido desarrollado con el objetivo de proporcionar una solución integral para el monitoreo de variables vitales y la gestión de medicamentos de manera no invasiva, facilitando así la vida diaria de los pacientes con problemas de salud. A lo largo del desarrollo, hemos logrado cumplir con los requisitos establecidos, ofreciendo un sistema competitivo en cuanto a funcionalidades y precio en comparación con otras alternativas disponibles en el mercado.

Sin embargo, es importante destacar que el sistema no está exento de limitaciones. La comunicación Wifi entre el brazalete y el pastillero tiene restricciones en cuanto a distancia, especialmente si no hay una red Wifi, lo que puede afectar la experiencia del usuario en ciertos entornos. Además, la duración de la batería del brazalete puede resultar insuficiente para algunos usuarios, requiriendo una recarga diaria.

En resumen, a pesar de las limitaciones identificadas, el proyecto ESPill representa un paso significativo hacia adelante en la mejora de la calidad de vida de los pacientes con problemas de salud.

9. Referencias

Alzheimers.gov/es. (20 de febrero de 2024). *¿Qué es la enfermedad de Alzheimer?* <https://www.alzheimers.gov/es/alzheimer-demencias/enfermedad-alzheimer>

Arduino ESP32 y Arduino ESP8266 series #8. *Noche de Programación*. <https://nocheprogramacion.com/series/arduino-esp/008-mdns-esp.html>

Concha, R. (2015). *Guía de iniciación a APP INVENTOR*. <https://codeweek.eu/docs/spain/guia-iniciacion-app-inventor.pdf>

Elegocart (s.f.). *Interfacing MAX30100 Heart Rate Monitor With Arduino*. *Autodesk Instructables*. <https://www.instructables.com/Interfacing-MAX30100-Heart-Rate-Monitor-With-Arduino/>

Gras, V. (18 de febrero de 2020). *BMP280. Sensor de presión atmosférica*. *TuriBOT. Electrónica y Robótica*. <https://www.turibot.es/blog/index.php/2020/02/18/bmp280-sensor-de-presion-atmosferica/>

Instituto Nacional de Estadísticas y Censos (INDEC) (2010). *Tablero de Personas Mayores*. <https://www.indec.gob.ar/indec/web/Nivel4-Tema-2-21-130>

Llamas, L. (1 de abril de 2020). *Medir temperatura y presión barométrica con Arduino y BMP280*. *Luis Llamas. Ingeniería, informática y diseño*. <https://www.luisllamas.es/medir-temperatura-y-presion-barometrica-con-arduino-y-bmp280/>

Robocraze (2023). *Interfacing MAX30100 Pulse Oximeter with Arduino*. <https://robocraze.com/blogs/post/how-to-use-max30100-arduino-as-heart-rate-sensor>

Santos, S. (s.f.). *ESP32 Web Bluetooth (BLE): Getting Started Guide*. *Random Nerd Tutorials*. <https://randomnerdtutorials.com/esp32-web-bluetooth/>

The MathWorks, Inc. (2024). *ThingSpeak. The IoT Platform with MATLAB Analytics*. https://www.mathworks.com/help/thingspeak/index.html?s_tid=CRUX_lftnav

Wojcik, S. y Kang, S. (2024). *Signos vitales (temperatura corporal, pulso, frecuencia respiratoria, presión arterial)*. *University of Rochester Medical Center Rochester*. <https://www.urmc.rochester.edu/encyclopedia/content.aspx?ContentTypeID=85&ContentID=P03963>

Zúñiga Domínguez, J. C. (s.f.). *No uses IP usa mDNS web server con ESP32 y ESP8266*

10. Bibliografía

- Alzheimers.gov/es. (20 de febrero de 2024). *¿Qué es la enfermedad de Alzheimer?* <https://www.alzheimers.gov/es/alzheimer-demencias/enfermedad-alzheimer>
- Concha, R. (2015). *Guía de iniciación a APP INVENTOR*. <https://codeweek.eu/docs/spain/guia-iniciacion-app-inventor.pdf>

- Elegocart (s.f.). Interfacing MAX30100 Heart Rate Monitor With Arduino. *Autodesk Instructables*. <https://www.instructables.com/Interfacing-MAX30100-Heart-Rate-Monitor-With-Ardui/>
- Gras, V. (18 de febrero de 2020). BMP280. Sensor de presión atmosférica. *TuriBOT. Electrónica y Robótica*. <https://www.turibot.es/blog/index.php/2020/02/18/bmp280-sensor-de-presion-atmosferica/>
- Instituto Nacional de Estadísticas y Censos (INDEC) (2010). *Tablero de Personas Mayores*. <https://www.indec.gob.ar/indec/web/Nivel4-Tema-2-21-130>.
- Llamas, L. (1 de abril de 2020). Medir temperatura y presión barométrica con Arduino y BMP280. *Luis Llamas. Ingeniería, informática y diseño*. <https://www.luisllamas.es/medir-temperatura-y-presion-barometrica-con-arduino-y-bmp280/>
- Robocraze (2023). Interfacing MAX30100 Pulse Oximeter with Arduino. <https://robocraze.com/blogs/post/how-to-use-max30100-arduino-as-heart-rate-sensor>
- Santos, S. (s.f.). ESP32 Web Bluetooth (BLE): Getting Started Guide. *Random Nerd Tutorials*. <https://randomnerdtutorials.com/esp32-web-bluetooth/>
- The MathWorks, Inc. (2024). *ThingSpeak. The IoT Platform with MATLAB Analytics*. https://www.mathworks.com/help/thingspeak/index.html?s_tid=CRUX_lftnav
- Wojcik, S. y Kang, S. (2024). *Signos vitales (temperatura corporal, pulso, frecuencia respiratoria, presión arterial)*. University of Rochester Medical Center Rochester. <https://www.urmc.rochester.edu/encyclopedia/content.aspx?ContentTypeID=85&ContentID=P03963>
- Zúñiga Domínguez, J. C. (s.f.). No uses IP usa mDNS web server con ESP32 y ESP8266 Arduino ESP32 y Arduino ESP8266 series #8. *Noche de Programación*. https://nocheprogramacion.com/series/arduino_esp/008_mdns_esp.html