



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE  
ESCUELA DE INGENIERÍA  
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

IEE2913 — Diseño Eléctrico

## Informe final

Grupo N°4: Adriana Navarrete - Guillermo Vidal - Diego Marusic  
12 de diciembre de 2020

---

---

### 0.1. Abstract

Due to the current pandemic, it has been necessary to free the largest number of beds for them to be used by seriously ill COVID patients. This leads to the problem of how to continue to monitor patients who are not critical but do need constant evaluation of their vital signs. From this problem arises the solution of designing a low-cost medical telemetry system, where it is possible to see the vital signs of non-critical patients from home, without the need for the doctor to visit patients. The information obtained from the patient and then sent to the doctor is: Electrocardiogram, respiratory rate, heart rate, oxygen saturation in the blood, body temperature and pupil reflex. In addition, the patient can enter information through audio and the pressing of buttons to indicate the performance of certain activities.

# **1. Descripción del proyecto**

El dispositivo creado posibilita ver los signos vitales de pacientes no críticos desde su domicilio. La información se adquiere con distintos sensores, se procesa, y luego, a través de un microcontrolador se envía a una interfaz personal a la que tendrán acceso médico y usuario. La información será recopilada con distintos sensores, específicamente, para el electrocardiograma se usarán 3 electrodos. Para la frecuencia respiratoria, se usará una mascarilla que tiene incorporado un sensor que detecta la respiración. Para la frecuencia cardíaca y la saturación de oxígeno en la sangre se utilizarán un sensor de pulso cardíaco y un oxímetro respectivamente. Para la temperatura corporal, se utilizará un sensor especial que debe estar en contacto con la axila de la persona. Finalmente, para el reflejo de pupilas se utilizará una cámara empotrada a la pared del dispositivo.

# **2. Requerimientos y especificaciones**

Como se mencionó anteriormente, el presente proyecto consiste en un sistema de telemedicina médica de bajo costo, que pueda recolectar información del estado de un paciente no crítico desde su domicilio. Estos datos deben ser procesados y enviados a un servidor para su posterior visualización por personal médico calificado. La interfaz con el usuario debe ser lo más cómoda posible, tanto para el paciente como para el médico.

Especificaciones:

La información a procesar son variables que serán medidas o estimadas a partir de sensores biométricos con el acondicionamiento analógico y/o digital necesario. Estas son:

- Electrocardiograma
- Frecuencia cardíaca
- Saturación de oxígeno en sangre
- Temperatura corporal

Excepto por el electrocardiograma, que debe ser visto por el médico tratante, la información debe ser actualizada cada 1 segundo.

Adicionalmente, el módulo incluirá la posibilidad de un examen de reflejo de pupilas, el cual debe ser realizado con los actuadores respectivos y adquirido con una cámara. El protocolo para este examen debe ser amigable con el paciente.

El paciente también debe informar a través del dispositivo, las horas cuando este come, orina, evacua o toma algún fármaco. Por otro lado, el paciente debe tener la oportunidad de

enviar un mensaje de audio reportando alguna anomalía.

El sistema debe contar con detección y alerta de errores o mal funcionamiento, indicando de manera simple y clara al paciente y al servidor de este hecho.

### 3. Prototipo Final

#### 3.1. Presentación del prototipo

Las siguientes imágenes muestran el prototipo final implementado.

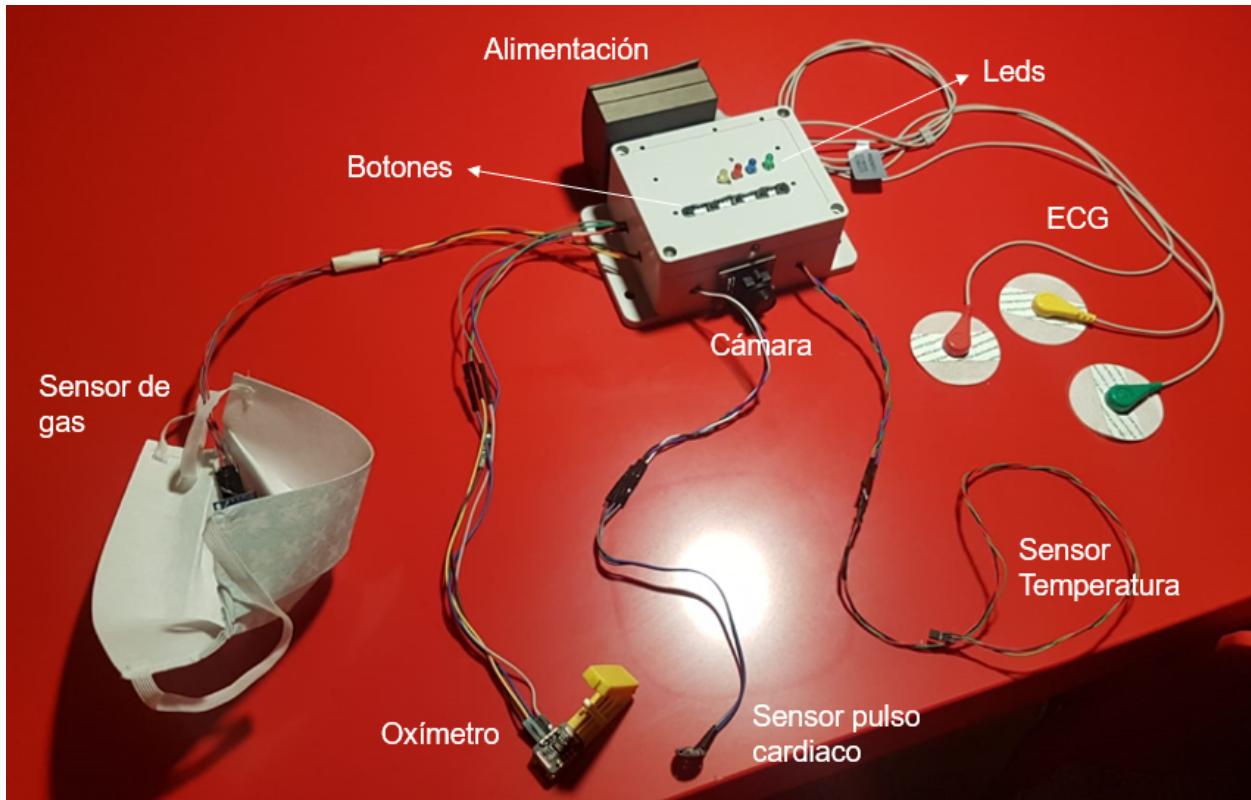


Figura 1: Prototipo final

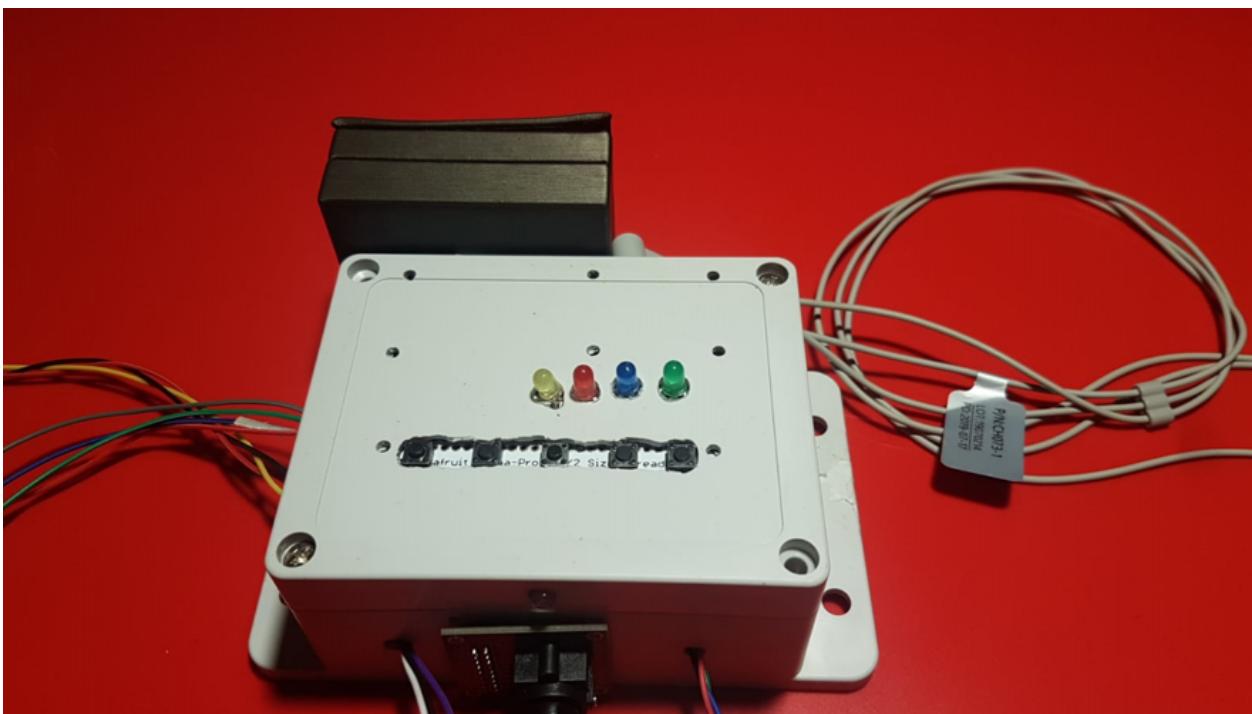


Figura 2: Prototipo final vista superior

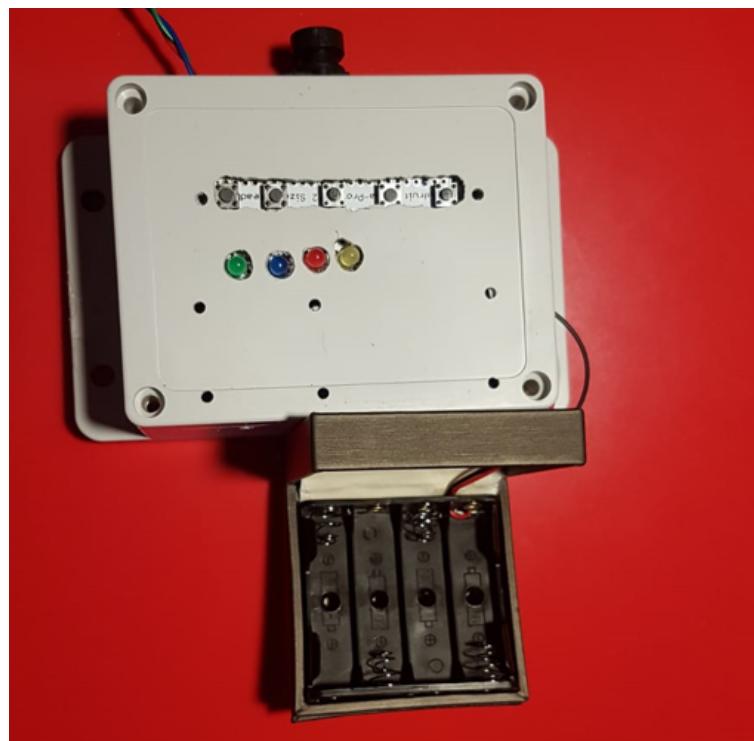


Figura 3: Prototipo final compartimiento de la alimentación

Como se puede ver en las imágenes anteriores, el dispositivo consiste en una caja rectangular gris con las puntas redondeadas, de la cual se desprenden los distintos sensores.

El electrocardgiograma se debe conectar como se muestra en la siguiente imagen respetando los colores. La descripción de como se ejecuta el examen se entrega en la sección 6.9.3 del presente informe.

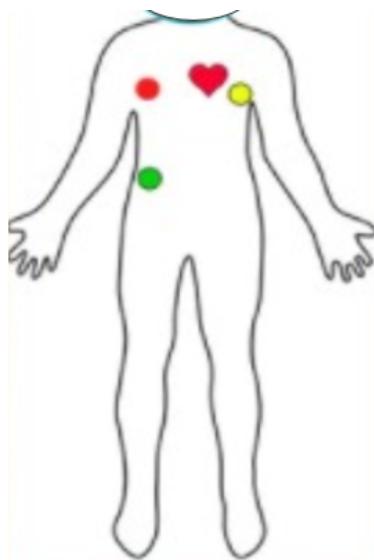


Figura 4: Conectar electrodos al cuerpo respetando los colores de la imagen

El sensor de temperatura se debe pegar con cinta adhesiva (no incluída) en la axila del paciente, de modo simulando el contacto de un termómetro normal

El sensor de pulso cardiaco y el oxímetro traen pinzas que se deben colocar en los pulgares (independiente de izquierda o derecha)

La mascarilla KN95 se debe colocar como una mascarilla normal y se debe respirar normalmente

Hay 5 botones en la sección superior del prototipo, considerando que de derecha (led verde) a izquierda (led amarillo) se enumeran del 1 al 5, se deben apretar en el siguiente orden al momento de realizar cualquiera de las siguientes 4 actividades: Orinar (botón 1), defecar (botón 2), comer (botón 3) e ingerir fármacos (botón 4). Al apretar un botón se prenderá el led azul que representa que el reporte de actividad se realizó correctamente. Además, el quinto botón se debe apretar al momento de querer realizar streaming con la cámara cuando el médico así lo solicite. Por otra parte, dichos leds también se pueden encender para solicitar exámenes o reportar errores. La siguiente imagen resume el significado cuando el paciente observe que se encienda un determinado led.



Figura 5: Significado de cada led

Finalmente, para alimentar el dispositivo se debe abrir la caja negra lateral y reemplazar las pilas.

### 3.2. Análisis de cumplimiento de especificaciones

En cuanto al cumplimiento de las especificaciones, se cumple la frecuencia de muestreo de cada sensor y los signos vitales se actualizan cada 1 segundo. Por su parte, el electrocardiograma, que debe ser solicitado por el doctor, se actualiza cada 1.2 segundos. Más aun, en la sección de implementación y resultados se puede ver en la interfaz implementada un electrocardiograma, un examen de frecuencia cardíaca que muestra entre 87 y 90 pulsaciones por minuto (figura 11), un examen de frecuencia respiratoria que muestra 16 respiraciones por minuto, un examen de saturación de oxígeno que muestra una saturación de 99 % y un examen de temperatura corporal que muestra una temperatura que varía entre 35.8 y 36.1 (dada la sensibilidad del sensor utilizado).

La figura 6 resume los valores obtenidos para los 4 exámenes anteriores en un determinado instante de tiempo, mientras que la figura 7 muestra cuales debiesen ser esos valores en un adulto normal en reposo.

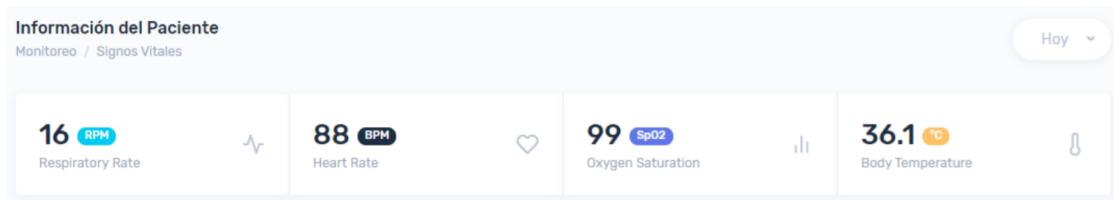


Figura 6: Valores de los signos vitales obtenidos

Valores Normales	Frecuencia Cardiaca	Frecuencia Respiratoria	Saturación Oxígeno	Temperatura
Niños (6-8 años)	100 - 115	20 - 25	95- 100	37.0
Adultos	60 – 80	15 – 20	95- 100	36.0 – 37.0

Figura 7: Valores de referencias signos vitales normales en reposo

En las imágenes anteriores se puede apreciar, que a excepción de la frecuencia cardiaca todos los valores están dentro del rango. La frecuencia cardiaca difiere levemente, porque al momento de realizar el examen la persona no estaba acostada y tenía más adrenalina de lo normal.

Además de lo anterior, se logró implementar correctamente el examen de reflejo de pupilas, mediante un streaming en que se puede apreciar correctamente la dilatación y contracción de las pupilas (figura 17).

Cabe destacar también, que el paciente es capaz de reportar las actividades solicitadas mediante la pulsación de un botón dependiendo de la actividad respectiva (figura 20). Además, la interfaz tiene habilitada una sección para informar anomalías por mensajes de audio (figura 31).

Finalmente, a modo de alertar de errores, en caso de que al momento de apretar un botón al realizar una actividad, se debe encender el led azul en la sección superior del dispositivo para notificar que dicho reporte se transmitió correctamente, de lo contrario no se ha transmitido el reporte de dicha actividad y el paciente debe apretar nuevamente. Por otra parte, cuando el oxímetro presenta fallas ya sea porque se produjo una lectura erronea o falla la alimentación, se enciende el led amarillo de la sección superior del dispositivo para notificar al paciente, y, para notificar al médico se muestra una lectura de -999.

### 3.3. Bill of Materials

La siguiente tabla resume los costos de los materiales utilizados para realizar el prototipo final donde el precio fue determinado en pesos chilenos con el valor del dolar correspondiente al 6/12/2020, es decir, 1 dolar equivale a 744 pesos chilenos:

Producto	Precio (IVA Incluido)	Disponible en
Caja Proyectos Electrónicos 115x90x55mm	9990	maxelectronica
Cable Plano Arcoiris 20 Vías AWG28 20P 1 Metro	3490	maxelectronica
Plug Conector de Batería 9V para Jack Arduino	690	maxelectronica
Módulo Terminal con Borneras Tipo Bloque	1245	maxelectronica
Modelo KF301-2P 5 unidades		
BATERIA ALCAL 9V ENERG 522-BAT	3.990	Sodimac
MASCARILLA KN95	2.490	Sodimac
ADHESIVO POXIPOL TRANSP 14MADH	4390	Sodimac
PERNO 1/8X3/4+TUECRA+GOLIL PER	1490	Sodimac
Placa DOIT WROOM ESP32 Bluetooth + WiFi	9000	Afel
Sensor Pulso Cardiaco Corazón	3500	Afel
Conecotor terminal 3 pines con tornillo KF301-3P	300	Afel
Sensor Calidad Aire MQ135	3400	Afel
Camara OV7670 para Arduino	4500	Afel
Kit Boton Tactil 6x6x5 - Interruptor - 5 unidades	500	Afel
Pack 60 Cables de Conexion	3990	Afel
Sensor de Frecuencia Cardiaca ECG AD8232 Electrocardiograma	9990	Afel
Sensor Pulsioximetría Max30102	5990	Afel
Soporte de acrílico para Camara OV7670	1000	Afel
ESP-WROOM-32	5952	Mouser
Cable USB	2894	Mouser
Sensor de temperatura lm35	1600	mercadolibre
Regulador de tensión lm317 2 unidades	1000	mercadolibre
Impresión PCB	1000	mercadolibre
Amplificador operacional LM324	750	electronicaplugandplay
Capacitores, resistencias y diodos varios	2000	estimación
<b>TOTAL</b>	<b>78.667</b>	

Figura 8: Costos de los materiales para realizar el prototipo final

Cabe destacar que descontando el IVA y sin considerar costos de envío, el costo de los materiales del proyecto es de 63.720 pesos chilenos. Más aún, considerando que la caja utilizada puede ser reemplazada por una impresión 3D como se mostrará en la sección “Implementación y resultados”, la cual tiene un costo de 3000 pesos chilenos por unidad para producción a escala, y descontando la batería que puede ser proporcionada por el usuario, el costo de los materiales para producir el proyecto puede llegar a ser 53.000 pesos chilenos.

## 4. Diagrama de bloques de alto nivel

La siguiente figura, muestra el diagrama de bloques de alto nivel que representa al prototipo final implementado:

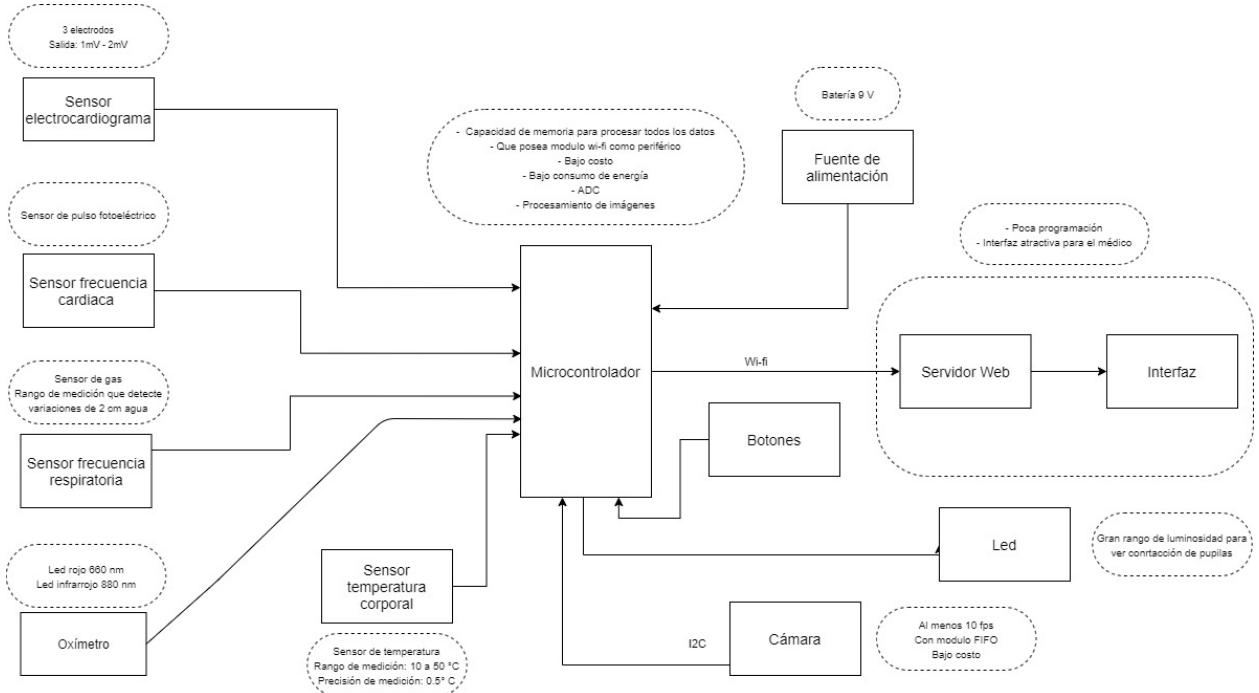


Figura 9: Diagrama de bloques de alto nivel

Como se puede apreciar en el diagrama anterior, este consiste de 5 sensores (uno para cada variable) y una cámara, que se comunican con el microcontrolador, el cual es el encargado de enviar la información al servidor para que se despliegue en la interfaz. Al mismo tiempo, habrán botones y leds para facilitar la comunicación que requerirá la solución implementada.

## 5. Diagrama de bloques de bajo nivel

La siguiente figura, muestra el diagrama de bloques de bajo nivel que representa al prototipo final implementado:

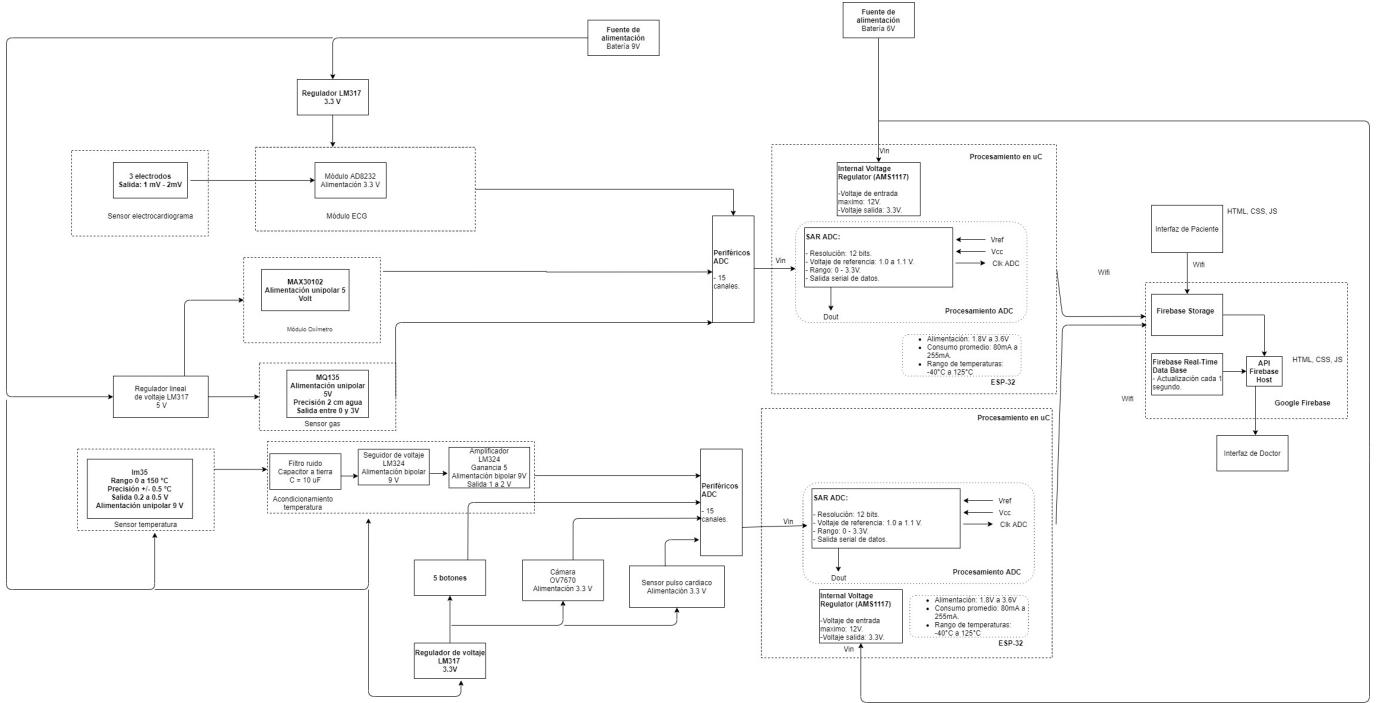


Figura 10: Diagrama de bloques de bajo nivel

Como se puede apreciar en el diagrama anterior, se utilizaron dos microncontroladores ESP32 producto de la falta pines para conectar todo a un solo microcontrolador. Ambos microcontroladores están conectados a una batería de 6 Volts compuesta or 4 pilas de 1.5 Volts en serie para suplirlas de la corriente que necesitan. A la ESP32 del bloque superior, se conectaron el electrocardiograma (módulo AD8232), el oxímetro (módulo MAX30102) y el sensor de gas (sensor MQ135). Tanto el oxímetro como el sensor de gas se alimentan con 5V, mientras que el electrocardiograma se alimenta a 3.3V. Los voltajes anteriores se obtienen de la salida de dos reguladores LM317 que son alimentados mediante una batería de 9V.

Por su parte, a la ESP32 del bloque inferior se conectan la cámara OV7670, el sensor de pulso cardiaco y los botones para reportar las actividades realizadas. Todos los componentes anteriores se alimentan con 3.3V que se obtienen a la salida del regulador lineal LM317, el cual es alimentado con la batería de 9V. Al mismo tiempo, a dicha ESP32 se conecta el sensor de temperatura LM35 (tras su posterior acondicionamiento), el cual es alimentado directamente desde la batería de 9V.

Finalmente, las dos ESP32 procesan la información y la envían mediante WiFi al servidor. En este caso se está utilizando Google Firebase, una plataforma para el desarrollo de aplicaciones web y aplicaciones móviles. En este contexto, la información enviada por las dos ESP se almacena en la base de datos Firebase Real Time Data Base, para su posterior lectura y despliegue en la interfaz y que se actualiza cada un segundo. Por último, destacar que como

se observa en la parte derecha del diagrama existe una diferenciación entre la interfaz del usuario y paciente. Ambas programadas dentro del mismo servidor. Respecto al paciente, la interfaz permite subir audios propios o grabarlos directamente en la página y subirlos, de modo tal de darle la posibilidad a este de enviar mensajes de audio al personal médico calificado.

## 6. Implementación y resultados

A continuación se resumen los distintos bloques implementados en el proceso de construcción del prototipo final:

### 6.1. Electrocardiograma

Se utilizó el módulo AD8232. Dicho módulo se alimenta con 3.3 Volts e incluye 3 electrodos los cuales pueden ser posicionados en los dos brazos y el tobillo izquierdo o bien dos en el pecho y uno en costado derecho del abdomen. En particular, se consideraron 255 muestras por segundo, las cuales se guardan en un *array* que luego es enviado a la base de datos. La interfaz incorpora la visualización de 765 muestra lo que es equivalente a visualizar la lista de valores actuales junto con las dos mediciones anteriores. La página se actualiza cada 1.2 segundos. La siguiente imagen muestra la salida que se obtiene del electrocardiograma.

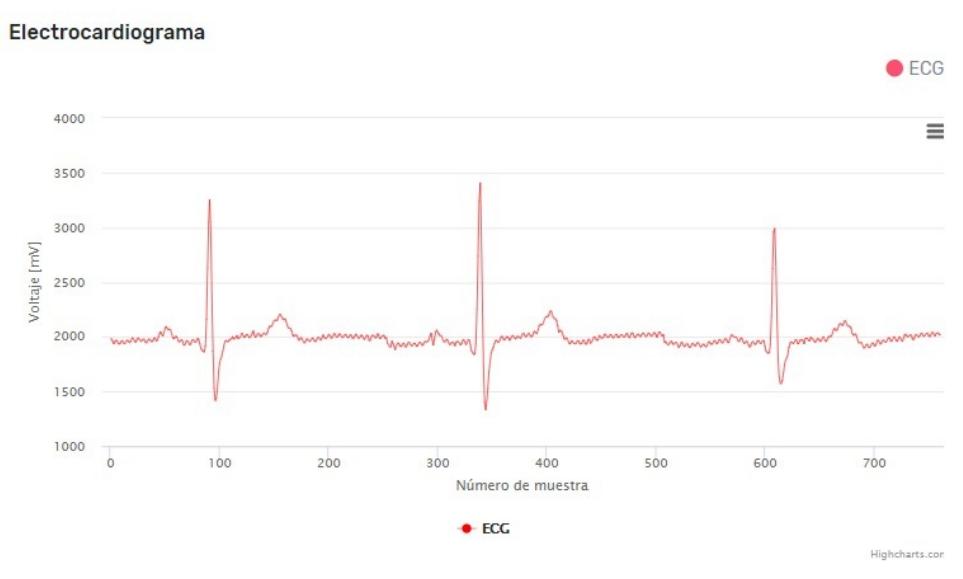


Figura 11: Electrocardiograma

## 6.2. Frecuencia cardíaca

Se utilizó un sensor de pulso cardiaco fotoeléctrico, el cual es alimentado con 3.3 Volts. La salida del sensor se muestra en la siguiente imagen:

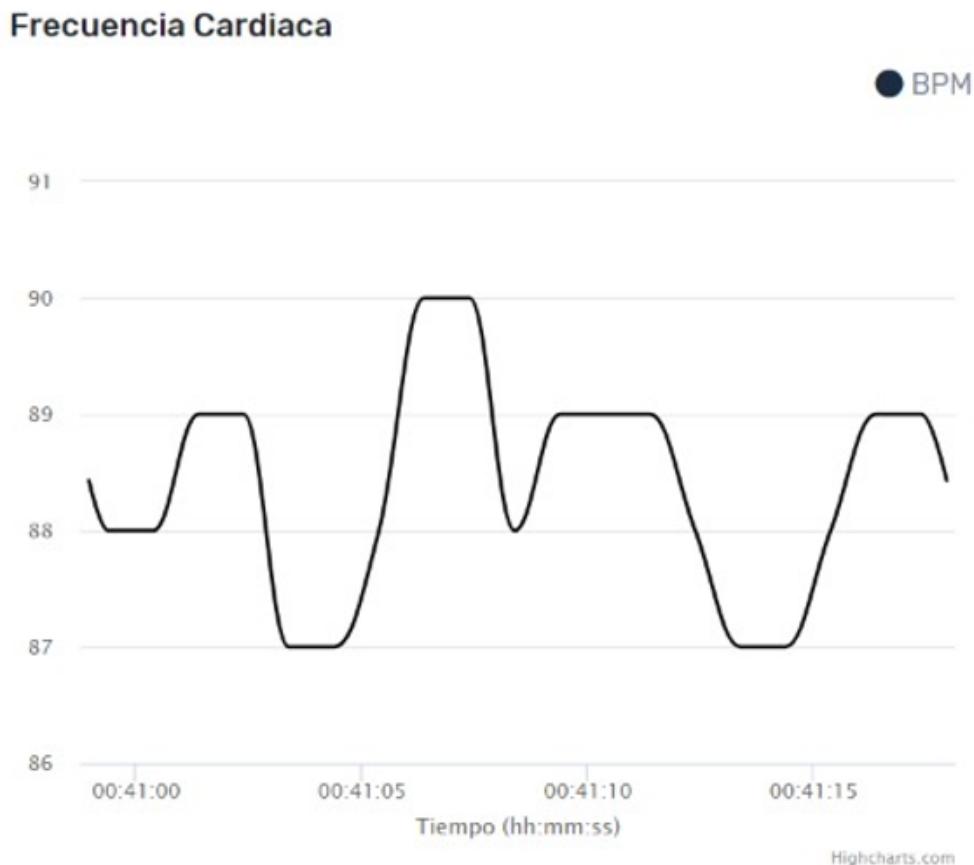


Figura 12: Frecuencia cardíaca

## 6.3. Frecuencia respiratoria

Se utilizó el sensor de gas MQ135 el cual se alimenta con 5 Volts y se adosó a una mascarilla KN95. El sensor varía su salida de voltaje dependiendo de la cantidad de CO<sub>2</sub> presente en el ambiente, por lo que en cada exhalación la salida alcanza un máximo, luego contando los máximos en un intervalo de tiempo se obtiene la frecuencia respiratoria. La siguiente imagen muestra la salida del sensor de gas implementado en donde en azul se ve como varía la concentración de dióxido de carbono con cada inhalación y exhalación.

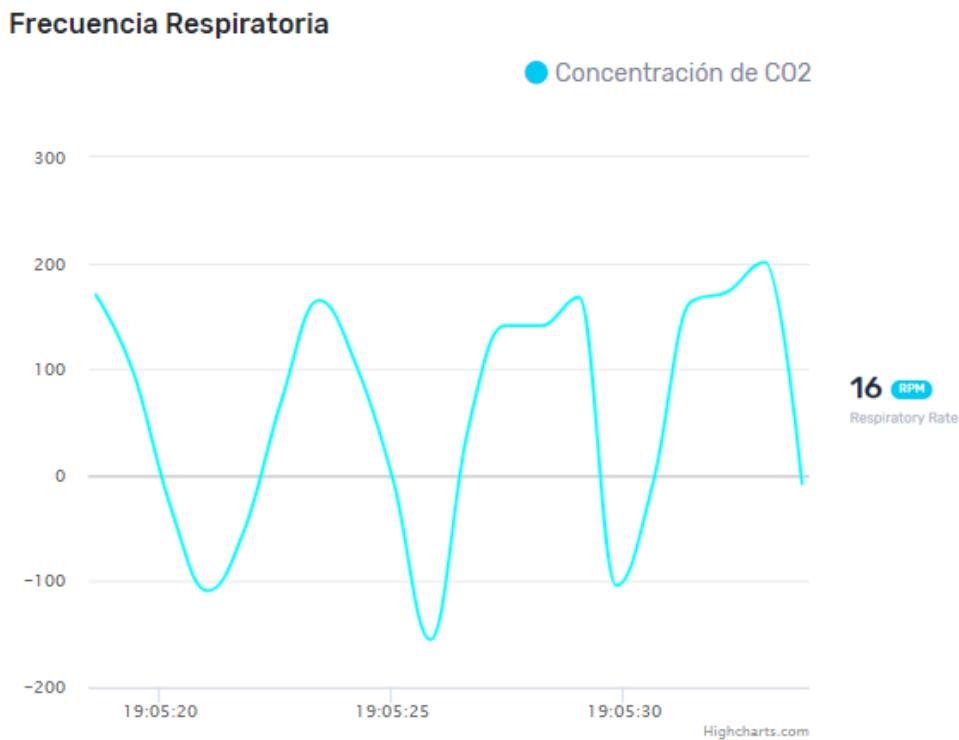


Figura 13: Frecuencia respiratoria

#### 6.4. Saturación de oxígeno en sangre

Se utilizó el módulo MAX30102 el cual se alimenta con 5 Volts. El módulo detecta la concentración de oxígeno en la sangre utilizando un led y un fototransistor, luego en base a la oposición del dedo a la luz que emite el led y recibe el fototransistor se determina la concentración anterior. La siguiente imagen muestra el funcionamiento del módulo MAX30102.

## Saturación de Oxígeno en la Sangre

● SpO2



Figura 14: Saturación de oxígeno en la sangre

## 6.5. Temperatura corporal

Se utilizó el sensor de temperatura lm35 y una etapa de acondicionamiento para obtener la ganancia que permite a la esp32 leer la salida del sensor. La siguiente figura muestra la simulación del circuito implementado donde la señal verde es la entrada y la azul la salida:

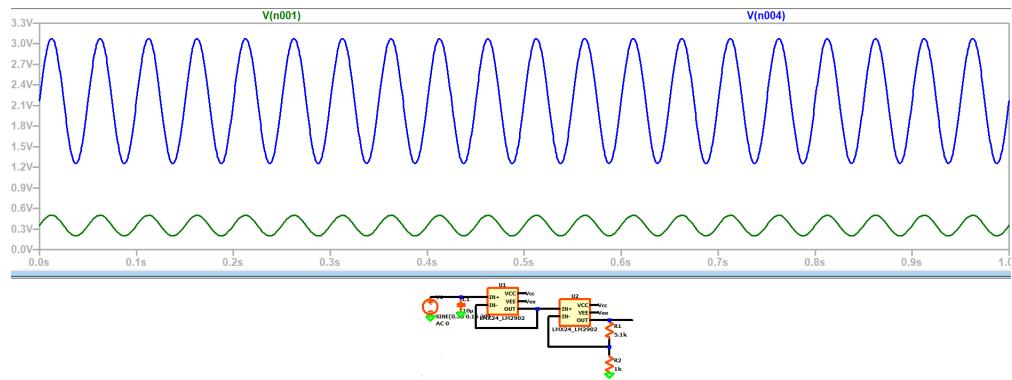


Figura 15: Simulación circuito temperatura

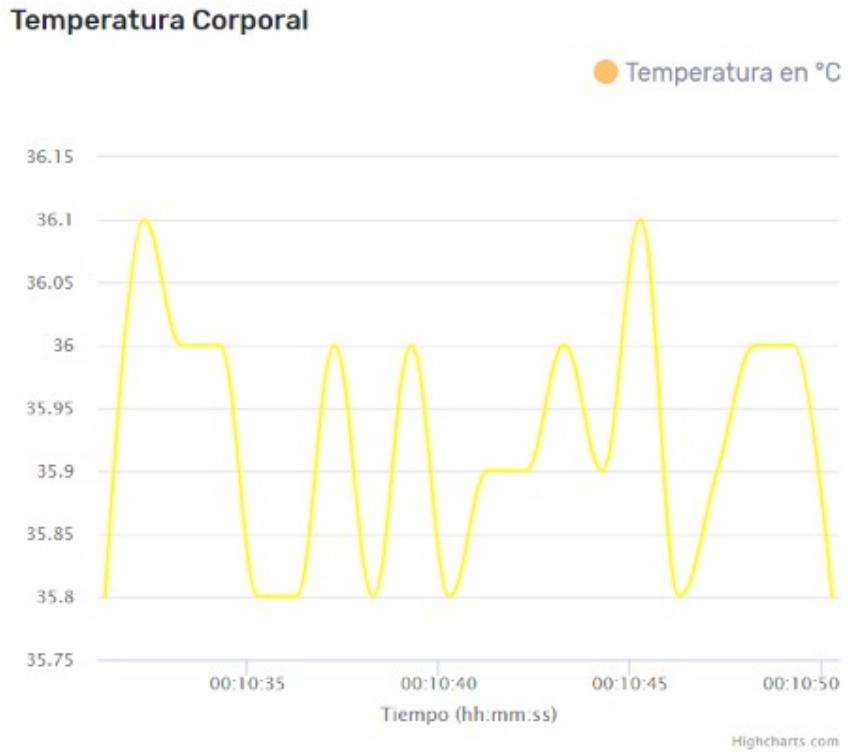


Figura 16: Resultados Temperatura Corporal

## 6.6. Examen de reflejo de pupilas

Se utilizó la cámara OV7670 la cual se alimenta con 3.3 V y se comunica mediante el protocolo I2C. En la configuración utilizada en el proyecto, esta se opera para grabar en tiempo real y realizar un *streaming* con un ciclo de trabajo máximo de 30 fotogramas por segundo (frames per second). Lo anterior, permite visualizar de manera continua los movimientos en tiempo real y en particular, la dilatación pupilar cuando se le agrega luz al ojo. Junto con esto la cámara también brinda la posibilidad de tomar fotografías en formato BMP para ser guardadas en por ejemplo, una tarjeta SD, sin embargo, en nuestro trabajo no se ha considerado.



Caso led apagado



Caso led encendido

Figura 17: Resultados Exámen de pupilas

## 6.7. Signos vitales

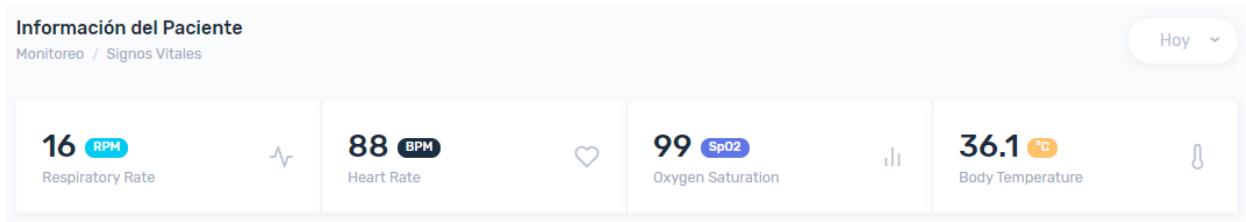


Figura 18: Valores obtenidos de signos vitales

## 6.8. Actividades

Evacuar				Orina			
#	Paciente	Actividad	Fecha	#	Paciente	Actividad	Fecha
1	Mark	Evacuar	12/12/2020, 1:18:50	1	Mark	Orinar	12/12/2020, 1:18:45
2	Mark	Evacuar	12/12/2020, 1:20:7	2	Mark	Orinar	12/12/2020, 1:20:2
3	Mark	Evacuar	12/12/2020, 1:21:11	3	Mark	Orinar	12/12/2020, 1:21:6
4	Mark			4	Mark		
5	Mark			5	Mark		
6	Mark			6	Mark		

Comida				Medicamentos			
#	Paciente	Actividad	Fecha	#	Paciente	Actividad	Fecha
1	Mark	Comer	12/12/2020, 1:18:41	1	Mark	Medicamentos	12/12/2020, 1:18:55
2	Mark	Comer	12/12/2020, 1:19:56	2	Mark	Medicamentos	12/12/2020, 1:20:15
3	Mark	Comer	12/12/2020, 1:21:3	3	Mark	Medicamentos	12/12/2020, 1:21:14
4	Mark			4	Mark		
5	Mark			5	Mark		

Figura 19: Tabla de actividades reportadas por el paciente

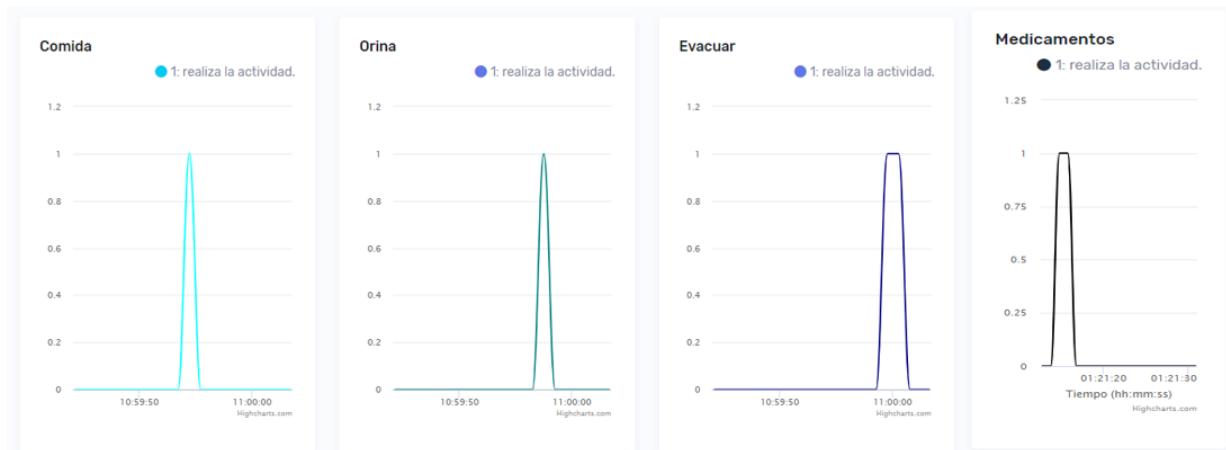


Figura 20: Gráficos en tiempo real del estado del botón de cada actividad

## 6.9. Reporte de Anomalías

### Historial

 Lista de Archivos

```
gs://testweb-f3370.appspot.com/Fri, 23 Oct 2020 15_37_21 GMT.wav
gs://testweb-f3370.appspot.com/Fri, 23 Oct 2020 18_23_15 GMT.wav
gs://testweb-f3370.appspot.com/Fri, 23 Oct 2020 19_01_11 GMT.wav
gs://testweb-f3370.appspot.com/Fri, 23 Oct 2020 19_31_58 GMT.wav
gs://testweb-f3370.appspot.com/Fri, 23 Oct 2020 21_50_23 GMT.wav
gs://testweb-f3370.appspot.com/Mon, 07 Dec 2020 16_09_10 GMT.wav
gs://testweb-f3370.appspot.com/Mon, 07 Dec 2020 16_28_11 GMT.wav
gs://testweb-f3370.appspot.com/Mon, 26 Oct 2020 00_25_58 GMT.wav
gs://testweb-f3370.appspot.com/Mon, 26 Oct 2020 08_21_38 GMT.wav
gs://testweb-f3370.appspot.com/Thu, 10 Dec 2020 12_47_10 GMT.wav
gs://testweb-f3370.appspot.com/Thu, 10 Dec 2020 15_47_24 GMT.wav
gs://testweb-f3370.appspot.com/Thu, 10 Dec 2020 15_51_08 GMT.wav
gs://testweb-f3370.appspot.com/Thu, 19 Nov 2020 01_13_45 GMT.wav
gs://testweb-f3370.appspot.com/Thu, 19 Nov 2020 01_18_45 GMT.wav
gs://testweb-f3370.appspot.com/Thu, 19 Nov 2020 12_15_15 GMT.wav
gs://testweb-f3370.appspot.com/Wed, 09 Dec 2020 20_36_23 GMT.wav
gs://testweb-f3370.appspot.com/WhatsApp Ptt 2020-10-24 at 16.38.33.ogg
```

Figura 21: Lista de archivos de audio reportados según fecha

## 6.10. Alimentación

El prototipo es alimentado por dos baterías, de 9 y 6 Volts. La batería de 6 Volts se usa únicamente para alimentar las dos ESP32. La batería de 9 Volts alimenta directamente al circuito de temperatura y a un regulador lineal de voltaje lm317 que entrega 5 Volts a su salida. A su vez, el regulador de 5 Volts alimenta al sensor de gas, el oxímetro, y a un segundo regulado lm317 puesto en cascada con el primero, que entrega 3.3 Volts a su salida y se encarga de alimentar al módulo del electrocardiograma (módulo AD8232), la cámara, el sensor de pulso cardiaco y los botones. Cabe destacar que los reguladores implementados no necesitan disipación térmica en este caso pero presentan una eficiencia de 55 % el primero y 66 % el segundo, por lo que en caso de realizar este diseño se recomienda fuertemente reemplazarlos por el módulo LTM8020 el cual para un proyecto de estas características presenta una eficiencia superior al 80 %.

## 6.11. Esquemático y PCB

La siguiente figura muestra el esquemático implementado para alimentar el prototipo y obtener las señales deseadas:

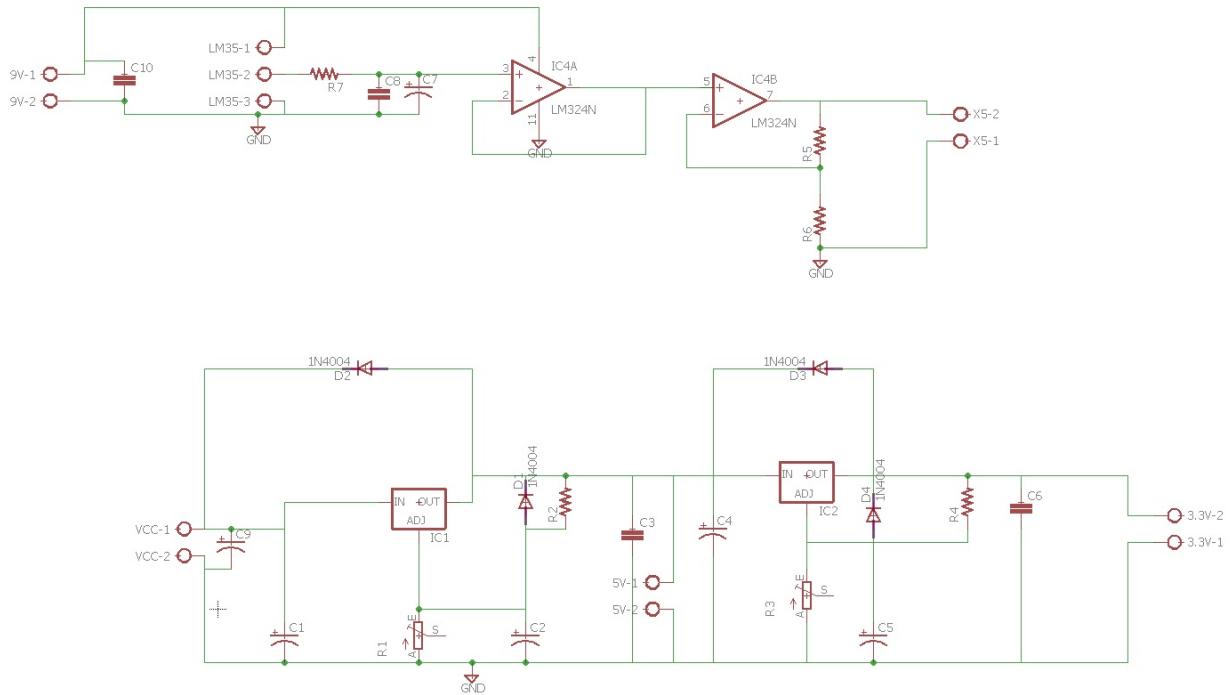


Figura 22: Esquemático prototipo final

En el esquemático anterior, la sección superior corresponde al circuito de temperatura, mientras que la inferior a los dos reguladores LM317 en casacada de los cuales se obtendrán los 5V y 3.3 V para alimentar al prototipo.

A continuación, la figura x corresponde al diseño de dos capas de la placa implementada, mientras que la figura y corresponde al diseño de una capa que se imprimió en la Universidad en caso de que el primer diseño no funcionase.

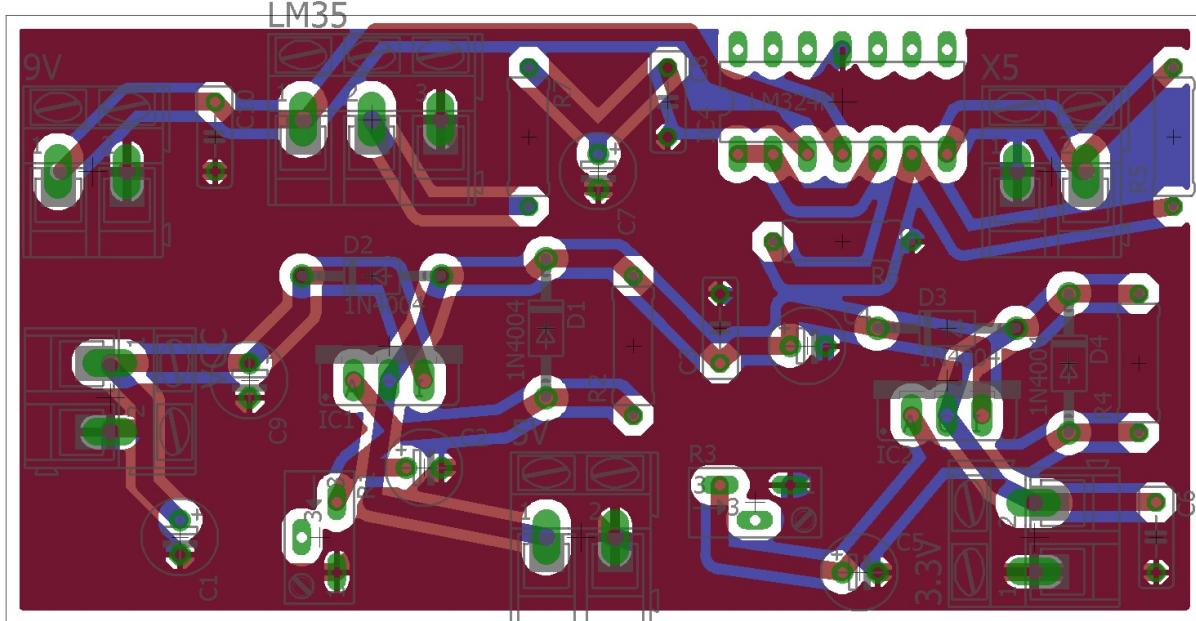


Figura 23: Diseño PCB dos capas

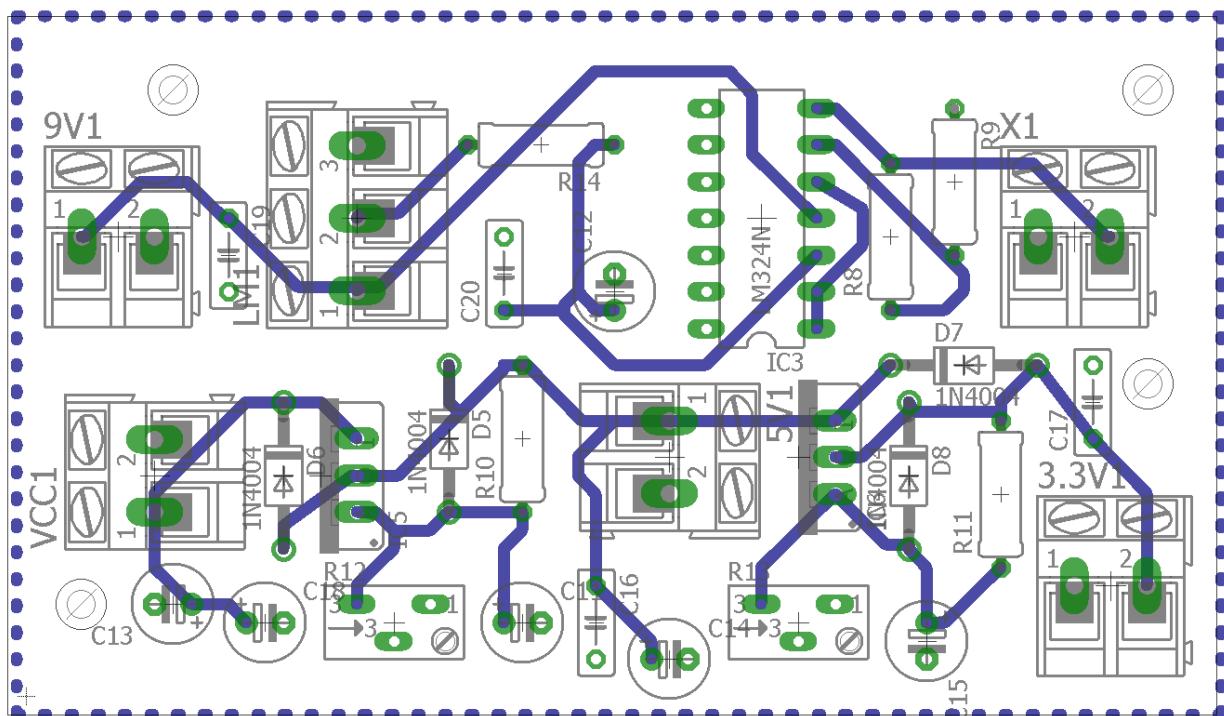


Figura 24: Diseño PCB una capa

## **6.12. Interfaz**

Para la creación de la interfaz tanto del doctor como del paciente, se ha programado un Host basado en códigos HTML, Javascript y CSS. En particular, para la creación de gráficos en tiempo real se han utilizado librerías de Highcharts que proveen código (HTML, JS y CSS) de manera gratuita para distintos tipos de gráficos. Luego, para el diseño del entorno web, se han descargado ejemplos de páginas web proveídas por Colorlib, las cuales han sido modificadas para adaptarse al formato del monitoreo remoto de pacientes. En este contexto, toda la información mostrada en la interfaz proviene de la base de datos en tiempo real de Google Firebase la que a su vez, recibe la información del dispositivo y cuyo tiempo de actualización es cada 1 segundo. Por último, para subir el host local a un entorno global de manera de ser visualizado desde cualquier dispositivo, se utiliza Google Firebase.

A continuación, se describe el entorno web programado mencionando las principales características que posee.

### **6.12.1. Login**

Con el objetivo de brindar confiabilidad y seguridad a la información transmitida por el paciente y al monitoreo remoto por parte del doctor se ha incorporado como página de inicio un formulario donde se piden las credenciales del usuario, es decir, el nombre del usuario y una contraseña asignada. Lo anterior, puede observarse en la Fig.25, donde existe un inicio de sesión tanto para el doctor como para el paciente.

### **6.12.2. Signos vitales**

Una vez de iniciar sesión en la ventana del doctor, se observará un panel de información donde se presentan los 4 signos vitales de interés: frecuencia respiratoria, frecuencia cardiaca, saturación de oxígeno en la sangre y la temperatura corporal. Junto con esto, se agregan gráficos en tiempo real donde se pueden observar los cambios que presentan dichas señales en el tiempo. Lo anterior, puede observarse en detalle en la Fig. 26.

### **6.12.3. Exámen electrocardiograma**

En adición a lo anterior, al costado izquierdo de la interfaz médica se encuentra una barra de tareas donde se pueden explorar y seleccionar la información que se deseé visualizar. En particular, en la sección *Exámenes/Ver/Electrocardiograma*, se encuentra la página donde se visualizarán los resultados asociados al exámen de electrocardiograma (Fig. 27) los cuales son actualizados cada 1.2 segundos. Junto con esto, el doctor tiene la posibilidad de solicitar el exámen del electrocardiograma o darlo por terminando apretando los dos botones de la parte izquierda los que enviarán la respectivas señales al paciente.

Por último, como información relevante, se incorpora el nombre del sensor utilizado y el periodo de muestreo empleado.

#### **6.12.4. Exámen reflejo pupilar**

De manera análoga al exámen de electrocardiograma, en la sección *Exámenes/Ver/Exámen de Pupilas*, se muestra la interfaz correspondiente a este exámen la cual se encuentra representada en la Fig. 28. En particular se sigue el mismo procedimiento anterior donde el doctor puede solicitar o terminar con el exámen enviando las señales correspondientes al paciente y en donde también puede manipular el actuador que permite encender y apagar el led de tal forma de observar la contracción de la pupila. En este contexto, para visualizar al paciente durante el exámen el médico debe presionar el botón asociado a la item *Streaming* donde se desplegará la página para comenzar la visualización en vivo.

Por último, como información relevante se incorpora el modo de operación de la cámara junto con el tipo de cámara empleada.

#### **6.12.5. Reporte de actividades**

Para visualizar los reportes de las actividades de ingerir medicamentos, comer, orinar y evacuar, el médico puede acceder a *Reportes/Mensajería/Actividades* donde encontrará una tabla asociada a cada actividad y donde podrá visualizar la fecha en la que fue realizada (Fig. 29). En específico, esta muestra el día, mes y año junto con la hora, minutos y segundos en la cual fue realizada y permite ir añadiendo en una nueva fila la información a medida que es recibida. A su vez, también existe un gráfico asociado a cada item de tal forma que se pueden ver en tiempo real los instantes donde se reportan las actividades.

#### **6.12.6. Reporte de anomalías**

Por último, en la sección *Reportes/Mensajería/Anomalías* el médico encontrará un botón donde podrá desplegar una lista de audios enviados por el paciente reportando sus anomalías. Para escucharlos, la página cuenta con un reproductor de audio el cual ingresando el nombre del audio, reproduce el contenido asociado. Además, en esta se cuenta con un buzón para el doctor en caso de ser oportuno, pueda incorporar archivos asociados al paciente y guardarlos en la nube. Lo anterior puede verse en más detalle en la Fig. 30.

#### **6.12.7. Interfaz paciente**

En caso de iniciar sesión como paciente, se mostrará un panel de instrucciones que deberá llevar a cabo el paciente para subir sus reportes de anomalías. En particular se incorpora la posibilidad de grabar el reporte en la misma interfaz y luego subir este mediante el buzón creado como se muestra en la Fig. 31.



Figura 25: Página de inicio de la interfaz

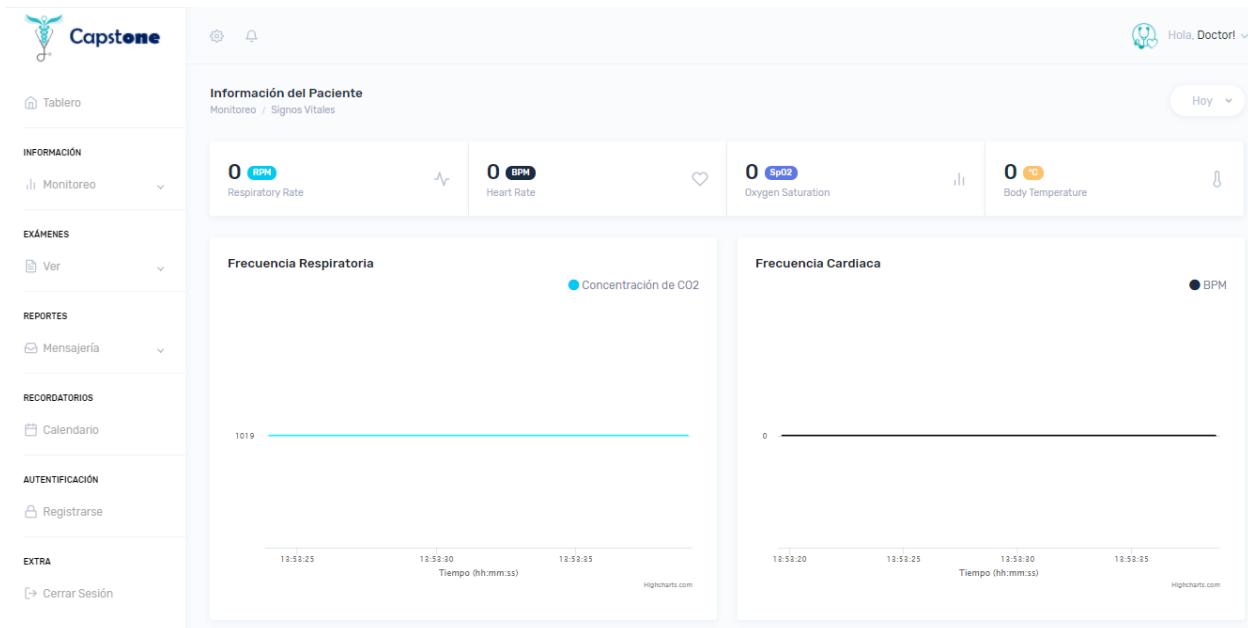


Figura 26: Página visualización de signos vitales



Figura 27: Página visualización electrocardiograma

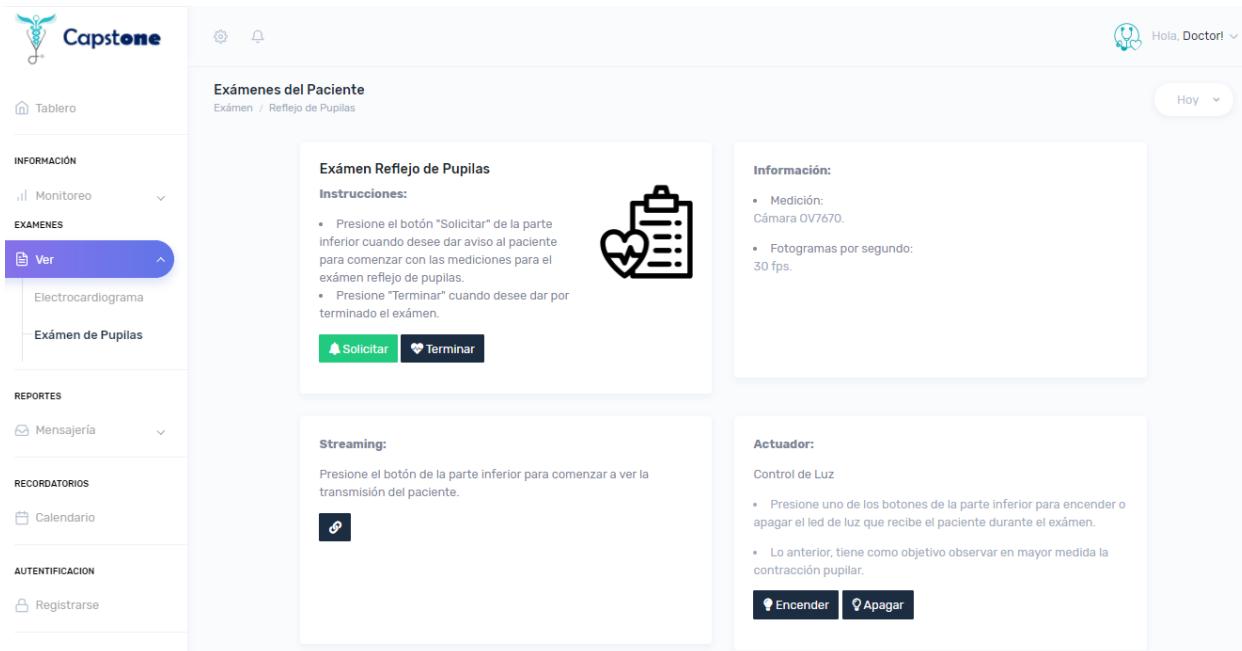


Figura 28: Página exámen pupilar

The screenshot shows the Capstone application's dashboard on the left and the 'Actividades del Paciente' (Patient Activities) report page on the right.

**Left Sidebar:**

- Logo: Capstone
- Tablero
- INFORMACIÓN
  - Monitoreo
- EXAMENES
  - Ver
- REPORTES
  - Mensajería
  - Actividades
  - Anomalías
- RECORDATORIOS
  - Calendario
- AUTENTIFICACION
  - Registrarse
- EXTRAS
  - Cerrar Sesión

**Right Page: Actividades del Paciente**

**Historial de reporte de actividades:**

A continuación, se reportan las actividades del paciente junto con el horario en que esta fue realizada.

#	Paciente	Actividad	Fecha
1	Mark		
2	Mark		
3	Mark		
4	Mark		
5	Mark		

**Medicamentos:**

1: realiza la actividad.

0

14:15:40 14:15:50  
Tiempo (hh:mm:ss) Highcharts.com

**Comida:**

#	Paciente	Actividad	Fecha
1	Mark		

1: realiza la actividad.

Figura 29: Página reporte de actividades

The screenshot shows the Capstone application's dashboard on the left and the 'Reportes del Paciente' (Patient Reports) page on the right.

**Left Sidebar:**

- Logo: Capstone
- Tablero
- INFORMACIÓN
  - Monitoreo
- EXAMENES
  - Ver
- REPORTES
  - Mensajería
  - Actividades
  - Anomalías
- RECORDATORIOS
  - Calendario

**Right Page: Reportes del Paciente**

**Reproductor**

**Instrucciones:**

- En lista de archivos encontrará el historial de reportes de audio enviados por el paciente. Haga click para desplegar el contenido.
- Copie sólo desde la fecha del audio que deseé escuchar y péguelo en la barra inferior según el formato mostrado.
- Haga click en reproducir.
- Si desea subir archivos adicionales al historial del paciente, utilice el buzón.

hello.wav

◀ ▶ ⏪

**Historial**

**Lista de Archivos**

Elegir archivos No se ha seleccionado ningún archivo

Subir

Figura 30: Página reporte de anomalías

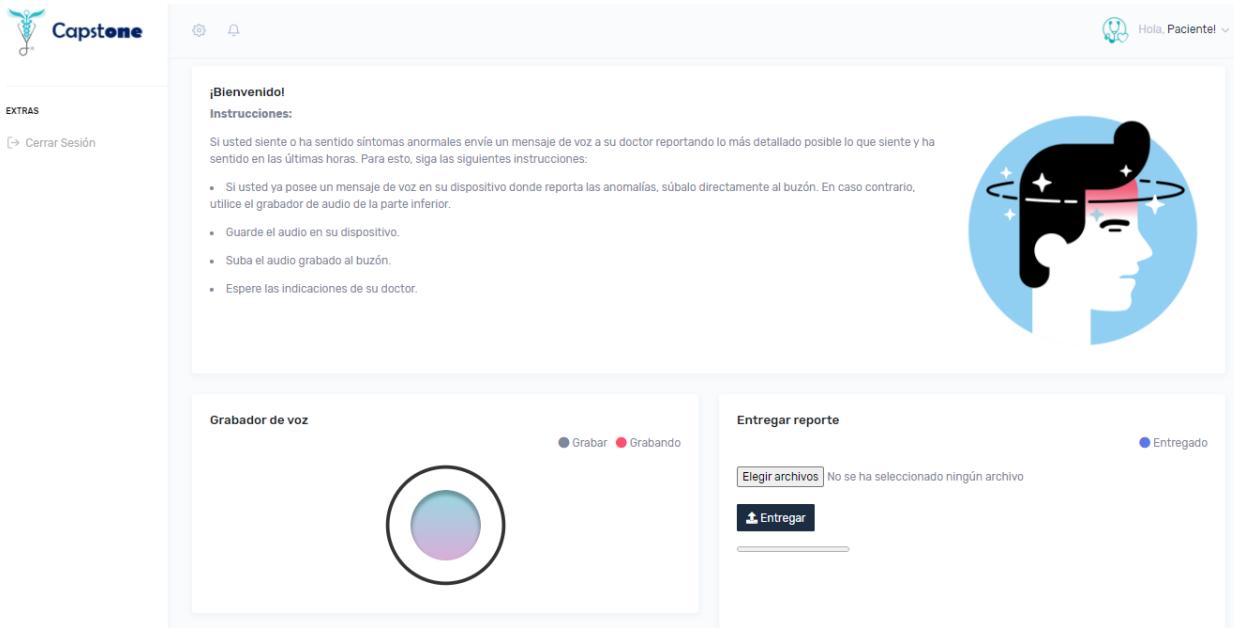


Figura 31: Buzón de reportes de anomalías

### 6.13. Diseño 3D

Finalmente, con el fin de darle más flexibilidad al diseño del prototipo y disminuir su costo, se propone un diseño 3D en caso de que se desee hacer una producción a escala del prototipo implementado.

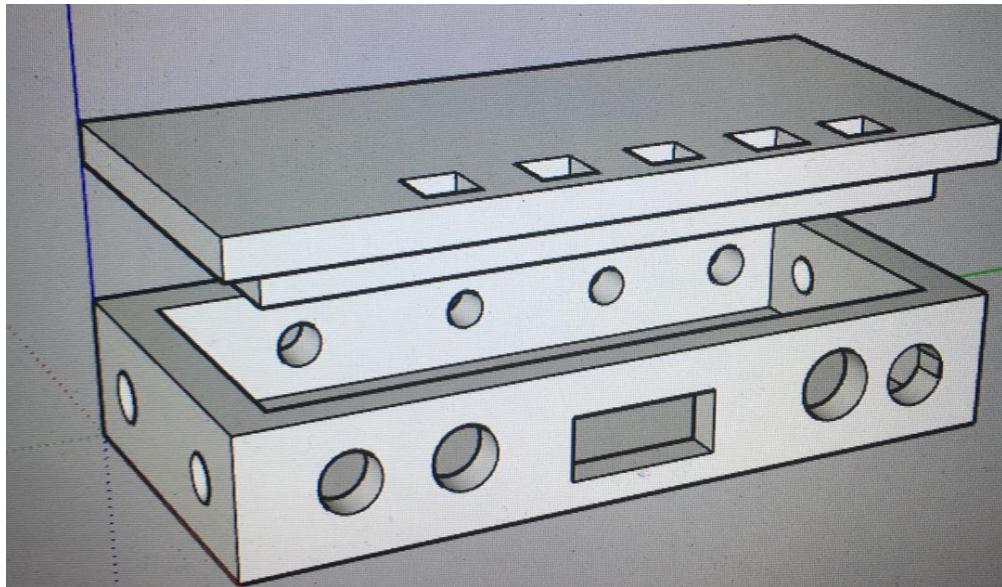


Figura 32: Diseño 3D del prototipo. Fuente: Elaboración propia

Como se puede ver en la imagen anterior, la cámara va empotrada en el rectángulo de la pared lateral, mientras que los círculos de las paredes laterales funcionan como ventilación y para sacar los cables necesarios al exterior de la caja. Por su parte el techo de la caja tiene 5 orificios donde irían los botones.

## 7. Cumplimiento de Estándares ECMA-287

Debido al creciente desarrollo de productos tecnológicos cada vez más interconectados, junto con la disminución a los costos asociados a ciertas herramientas tecnológicas, ha posibilitado que el diseño de estos, sea cada vez más accesible a nuevos desarrolladores. Bajo este contexto, surge la necesidad de normar y fijar estándares que salvaguarden la seguridad del usuario a modo de fijar los requisitos que deben cumplir los diseñadores de productos tecnológicos eléctricos. La importancia de considerar variables de ingeniería en el desarrollo de estos productos, radica en que por ejemplo, una descarga eléctrica en el cuerpo humano puede provocar efectos patofisiológicos peligrosos que pueden atentar con la vida de la persona como un paro cardiaco o un paro respiratorio. En consecuencia a lo anterior, y en concordancia al proyecto de telemetría médica, se analizará el cumplimiento del estándar de seguridad ECMA - 287.

### 7.1. Análisis de fuentes de energía

La norma ECMA-287, clasifica las fuentes de energía según los límites de tensión y corriente máximos que pueden brindar ante una carga resistiva considerando tanto situaciones normales como anormales (perturbaciones). En relación a esto, están las de tipo ES1, ES2 y ES3. La siguiente tabla muestra las principales características de cada una de estas considerando el tipo de alimentación.

	Energy source level		ES 1 limit	ES 2 limit	ES 3 limit
1a	DC	voltage	$\leq 60 \text{ V}$	$> 60 \text{ V} \leq 120 \text{ V}$	$> \text{ES 2 limit (figure 3.2)}$
1b		current	$\leq 2 \text{ mA}$	$> 2 \text{ mA} \leq 25 \text{ mA}$	
2a	AC (up to 1 kHz)	voltage	$\leq \text{ES 1 limit (figure 3.2)}$	$> \text{ES 1 limit (figure 3.2)}$	$> \text{ES 2 limit (figure 3.2)}$
2b		current	$\leq 0,5 \text{ mA r.m.s.}$	$> 0,5 \text{ mA} \leq 10 \text{ mA r.m.s.}$	
3a	combined AC and DC	voltage	$\frac{\text{Udc}}{60} + \frac{\text{Uac}}{\text{value below ES 1 limit (figure 3.2)}} \leq 1$	$\frac{\text{Udc}}{120} + \frac{\text{Uac}}{\text{value below ES 2 limit (figure 3.2)}} \leq 1$	

Figura 33: Fuente: ECMA - 287

El proyecto se alimenta en su totalidad por dos fuentes de alimentación. Una batería de 9 Volts y otra de 6 Volts compuesta por 4 pilas de 1.5 Volts puestas en serie. Dado que ambos voltajes están por debajo del límite de 60 Volts de las fuentes ES 1, se concluye que ambas fuentes de alimentación se clasifican como ES-1.

## 7.2. Análisis de protecciones, aislaciones y accesibilidad

Existen diferentes tipos de protecciones dependiendo del tipo de fuente de energía. Para nuestro caso particular al utilizar fuentes del tipo ES1, según la ECMA no es necesario interponer dispositivos protectores entre la fuente y una persona debido a que se asume que la impedancia del cuerpo es infinita. Pese a lo anterior, las dos fuentes de alimentación se encuentran dentro del cuerpo del dispositivo (para camabiarlas se debe abrir la tapa y cambiar las fuentes), de modo que se reduce al máximo la probabilidad de shock eléctrico. Además, al realizar la prueba del dedo, aplicando una fuerza de 10 N en todas las partes críticas del dispositivo, este aguantó correctamente sin desarmarse. Cabe destacar, que la caja utilizada para almacenar los circuitos es impermeable, por lo que también se encuentra protegido del agua.

Por su parte, el cable utilizado para realizar las conexiones entre dispositivos posee una tensión de aislamiento de 250 Volts, valor considerablemente mayor a los 9 Volts de la batería de mayor capacidad utilizada. Dado que se está trabajando con voltajes no superiores a 9 Volts, que se cuenta con cables que aislan más de 20 veces el voltaje anterior, y que las conexiones eléctricas del dispositivo, se encuentran además, aisladas por el cuerpo sólido e impermeable del mismo, se considera que se cumplen los estándares relacionadas a las aislaciones del ECMA - 287.

Dado, que se está trabajando con voltajes bajos, a excepción de los reguladores lineales, los componentes no están expuestos a tensiones que puedan ser dañinas. Sin embargo, los reguladores lineales utilizados podrían necesitar disipadores térmicos ya que se corre el riesgo de que alcancen una temperatura excesiva. Al hacer el análisis de disipación térmica, se observa que dichos reguladores alcanzarán una temperatura máxima de 80 grados, la cual es inferior a los 125 grados de temperatura máxima a la que pueden operar, por lo que se estima que no será necesario el uso de disipadores y se cumplirá el estándar referido a la seguridad de los componentes utilizados.

## 7.3. Riesgos mecánicos

La primera medida en que se incurrió para mitigar los riesgos mecánicos fue utilizar un cuerpo del dispositivo con todas sus esquinas redondas. De modo de minimizar el riesgo de herirse por golpearse con el dispositivo.

En segundo lugar todas las placas se encuentran atornilladas, a excepción de la PCB diseñada, la cual por error, se olvidó colocarle tornillos. En caso de repetir el diseño estos deben ser considerados. Sin embargo, se utilizó pegamento poxipol para adehrirla a la base del dispositivo por lo que no hay problemas para transportar el dispositivo de un lugar a otro.

Finalmente, todas las piezas sueltas como resistencias u otros dispositivos electronicos pequeños fueron soldados a las distintas placas y no se utilizaron protoboard en el diseño final, por lo que el dispositivo implementado es perctamente portable.

## 8. Análisis de Impacto de la Solución

Según el documento HEARTS Technical package for cardiovascular disease management in primary health care: access to essential medicines and technology publicado por la Organización Mundial de la Salud (2018), cada año mueren más personas por enfermedades cardiovasculares (ECV) que por cualquier otra causa. Además, se menciona que más de tres cuartas partes de las muertes relacionadas con cardiopatías y accidentes cerebrovasculares ocurren en países de ingresos medianos y bajos. Bajo este contexto, proyectos de telemetría médica que incluyan tanto un monitoreo de frecuencia cardiaca o un electrocardiograma y permitan el monitoreo remoto de pacientes mediante un kit de bajo costo toma suma importancia puesto que, puede posicionarse como una medida preventiva especialmente para los grupos de riesgo potencial o aquellas personas con bajos recursos económicos. Junto con esto, realizar un monitoreo de la respiración permitiría identificar y predecir eventos adversos causados por fallas respiratorias ya que un ritmo mayor a 27 rpm puede ser un indicador importante para predecir un paro cardíaco. De la misma manera, medir la temperatura corporal es de gran importancia puesto que ciertas enfermedades pueden controlarse monitoreando esta y la eficacia de un tratamiento iniciado puede ser evaluado por el médico.

En adición a lo anterior, y frente al complejo escenario que se vive en el mundo hoy producto de la pandemia causada por el virus COVID-19, el riesgo de asistir a centros de salud es cada vez mayor y puede arriesgar la salud de pacientes no contagiados. Mientras el llamado de las autoridades sigue siendo evitar el contacto estrecho y respetar las cuarentenas aún existen personas que deben asistir a hospitales públicos para ser atendidos o bien, monitoreados por diversas causas. A partir de esto, podemos observar el alto impacto social que tendría un sistema de monitoreo remoto de paciente, especialmente para aquellos estratos sociales más bajos. Debido a esto, es imperante considerar que las herramientas tecnológicas utilizadas para implementar la telemetría médica tengan los estándares suficientes para permitir al doctor realizar un correcto diagnóstico y asegurar la capacitación de este para su uso, es decir, deben ceñirse a una norma ética pues pueden impactar la directamente la salud de una persona. Además, junto a lo anterior, desde el punto de vista de diseño se debe asegurar que el producto cumpla con las normas de seguridad.

En relación al impacto económico y ambiental, para la elaboración del primer prototipo

se priorizó el factor económico antes que el ambiental. En particular, los reguladores empleados corresponden a reguladores lineales LM317 los cuales tienen una eficiencia para este caso de 55 % el primero y 66 % el segundo. La elección anterior se justifica ya que no fue necesario incurrir en costos adicionales porque dichos reguladores los proporcionó la Universidad. Sin embargo, el caso óptimo económico-ambiental hubiese sido comprar el módulo LTM8020 que presenta una eficiencia superior a 80 % y su implementación es extremadamente sencilla. Por otro lado, si bien actualmente se alimenta con una batería de 9 Volts y otra de 6 Volts, se considera que esta forma de alimentar no es la más eficiente. Lo ideal hubiese sido alimentar todo el dispositivo con 6 pilas de 1.5 Volts recargables en serie. De modo tal que como cada pila aportaría 2700 mAh, y el dispositivo consumiría 500 mAh, tendrían una duración de 32 horas. Con lo anterior, el dispositivo sería portable, de bajo impacto ambiental y eficiente en términos de alimentación (considerando el LTM8020 y un tiempo de duración de las baterías superior a un día). Cabe destacar, que el costo de producción del prototipo final fue inferior a 80.000 pesos chilenos, mientras que un dispositivo de similares características, y sin el examen de reflejo de pupilas, tiene un costo de venta de al menos 100.000 pesos chilenos, por lo que es viablemente económico entrar a competir en el mercado.

Finalmente, uno de los principales riesgos que podrían generarse con este tipo de tecnologías es el uso indebido por agentes no pertenecientes al área de la salud o bien, divulgación de información personal del paciente. A partir de esto, notamos oportuno al momento de realizar la interfaz incluir una sección de *login* y *register* con el objetivo de mitigar posibles filtraciones de información. Además, respecto al primer punto resulta difícil encontrar una medida para mitigarlo puesto que, escapa de nuestra área de control, sin embargo, una solución podría ser restringir la venta y uso del dispositivo a centros médicos con el fin de otorgar mayor confiabilidad al servicio.

## Referencias

- [1] <https://aprendiendoarduino.wordpress.com/2016/07/04/ethernet-shield/>
- [2] <https://www.web-robotica.com/arduino/como-funciona-el-modulo-arduino-ethernet-shield>
- [3] [https://naylampmechatronics.com/blog/17\\_Tutorial-M%C3%B3dulo-Ethernet-ENC28J60-y-Arduino.html](https://naylampmechatronics.com/blog/17_Tutorial-M%C3%B3dulo-Ethernet-ENC28J60-y-Arduino.html)
- [4] [https://www.naylampmechatronics.com/blog/12\\_Tutorial-B%C3%A1sico-de-Uso-del-M%C3%B3dulo-Bluetooth-H.html](https://www.naylampmechatronics.com/blog/12_Tutorial-B%C3%A1sico-de-Uso-del-M%C3%B3dulo-Bluetooth-H.html)
- [5] <https://programarfacil.com/podcast/esp8266-wifi-coste-arduino/>
- [6] <http://www.elhospital.com/temas/Analisis-de-la-frecuencia-de-muestreo-en-electrocardiografia+111828>
- [7] [http://www.scielo.org.mx/scielo.php?script=sci\\_arttext&pid=S1405-99402012000300001](http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1405-99402012000300001)
- [8] <http://dea.unsj.edu.ar/bioinstrumentacion2/diapecg.pdf>
- [9] <https://www.redalyc.org/pdf/304/30400405.pdf>
- [10] [https://link.springer.com/chapter/10.1007%2F978-3-540-74471-9\\_104](https://link.springer.com/chapter/10.1007%2F978-3-540-74471-9_104)
- [11] <https://www.pardell.es/tics,-telemedicina-y-esalud.html>
- [12] <http://red.uao.edu.co/bitstream/10614/12317/5/T09173.pdf>
- [13] <http://repository.udistrital.edu.co/bitstream/11349/13383/2/SarmientoG%C3%B3mezOscar2018.pdf>
- [14] <http://rinfo.fimdp.edu.ar/bitstream/handle/123456789/251/GDCometti%2BFCVogel-TFG-IEe-2018.pdf?sequence=1&isAllowed=y>

[15] <https://www.youtube.com/watch?v=YrwNc4Arsp0t=3s>

[16] [https://es.wikipedia.org/wiki/Ventilaci%C3%B3n\\_pulmonar](https://es.wikipedia.org/wiki/Ventilaci%C3%B3n_pulmonar): :text=Por %20convenio %20en %20el %20

[17] <https://www.energymed.com.co/estandar-de-seguridad-electrica-de-equipos-medico/>

[18] <https://www.youtube.com/watch?v=ZCr5G0zfnKY>

[19] <http://www.playbyte.es/electronica/tutoriales/modulo-wifi-esp-32/>

[20] <https://diyi0t.com/best-battery-for-esp32-nodemcu/>: :text=5V %20which %20is %20higher %20than, V

[21] <https://www.mouser.cl/ProductDetail/CUI-Inc/SWI10B-5-EW-I38?qs=sGAEpiMZZMvasLKgtn5bISc75q96P1ACJCAjt5gSs%252BrE7cgux0%252Bg4Q%3D%3D>

[22] <https://www.mouser.cl/ProductDetail/CUI-Inc/SWI10B-5-EW-I38?qs=sGAEpiMZZMvasLKgtn5bISc75q96P1ACJCAjt5gSs%252BrE7cgux0%252Bg4Q%3D%3D>

[23] <https://www.youtube.com/watch?v=gyzL-5M04OY>

[24] <https://www.youtube.com/watch?v=SLIdvurGt4Y>

[25] <https://www.youtube.com/watch?v=T65FQrvPpA8>

[26] <https://www.youtube.com/watch?v=msmdbm1V2PY>

[27] <https://www.youtube.com/watch?v=feudEXC0z9I>

[28] <https://swharden.com/wp/2013-06-10-precision-temperature-measurement/>

[29] <http://embedded-lab.com/blog/introducing-easy-pulse-a-diy-photoplethysmographic-sensor-for-measuring-heart-rate/>

- [30] [https://www.researchgate.net/figure/Ondas-de-senal-que-muestran-la-absolucion-de-luz-para-la-hemoglobina-oxigenada-y\\_fig4\\_331742636](https://www.researchgate.net/figure/Ondas-de-senal-que-muestran-la-absolucion-de-luz-para-la-hemoglobina-oxigenada-y_fig4_331742636)
- [31] <https://swharden.com/wp/2013-04-14-simple-diy-ecg-pulse-oximeter-version-2/>
- [32] <https://upcommons.upc.edu/bitstream/handle/2117/172341/Report.pdf?sequence=3isAllowed=y>
- [33] <https://www.youtube.com/watch?v=lxnNzrpLQOMt=445s>
- [34] <https://www.youtube.com/watch?v=RWEn0-fFWYwt=281s>
- [35] [https://www.youtube.com/watch?v=\\_twAIpQl93M](https://www.youtube.com/watch?v=_twAIpQl93M)
- [36] <https://www.youtube.com/watch?v=vUS-38ByYQUt=995s>
- [37] <http://www.openimpulse.com/blog/wp-content/uploads/wpsc/downloadables/OV7670-CMOS-VGA-Camera-Datasheet.pdf>
- [38] [https://www.youtube.com/watch?v=v\\_8XifFcpaI](https://www.youtube.com/watch?v=v_8XifFcpaI)
- [39] <https://www.youtube.com/watch?v=mtE7ppmfrxAt>
- [40] <https://hackaday.io/project/13156/logs?sort=oldest>

## 9. Anexos

```
main.cpp
src > main.cpp > ...
1 #include <Arduino.h>
2 #include <WiFi.h>
3 #include <FirebaseESP32.h>
4
5 #define FIREBASE_HOST "testweb-f3370.firebaseio.com"
6 #define FIREBASE_AUTH "FggzPj81zozI3XVvmH0Jn3aQM6vgGbU3Z00rZQb5c"
7 #define WIFI_SSID "ARRIS-6D22" //nombre de red wi-fi
8 #define WIFI_PASSWORD "Y43XTX9THAPTH4EF" //contraseña wi-fi
9
10 #define LED 13
11 #define Potenciometer 36
12 #define pulso_sensor 39
13 #define gas_sensor 34
14
15 TaskHandle_t Task1;
16
17 String path = "/Planificación II";
18 FirebaseDatabase firebaseData; //define un objeto de firebase
19
```

Figura 34: Anexo 1. Librerías, precompiler, creación de objeto en base de datos y framework Arduino.

```
//variables globales de sensado
float valor_sensado = 0;
float voltaje_sensado;

int pulso_sensado =0;

float gas_sensado = 0;
float breath =0;

bool enable_data = false;
```

Figura 35: Anexo 2. Variables globales utilizadas para el sensado

```
Serial.begin(9600);

//Inicialización conexión wi-fi
WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
Serial.print("Conectando a ....");
while (WiFi.status() != WL_CONNECTED){
| Serial.print(".");
| delay(300);
}

Serial.println(); Serial.print("Conectado con la IP:");
Serial.println(WiFi.localIP()); Serial.println();
```

Figura 36: Anexo 3. Setup: Inicialización de conexión wi-fi.

```

//Inicializacion Firebase
Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
Firebase.reconnectWiFi(true);
Firebase.setReadTimeout(firebaseData, 1000); //tiempo de espera de lectura de la base de datos
Firebase.setWriteSizeLimit(firebaseData, "tiny"); //tamaño y tiempo de escritura (tiny = 1s)

//definicion de variables en BDD: //para leer valores en bdd se ocupa get.Float
Firebase.setFloat(firebaseData, path + "/Sensores/Temperatura", 0.0);
Firebase.setFloat(firebaseData, path + "/Sensores/Sat_Oxigeno", 0.0);
Firebase.setFloat(firebaseData, path + "/Sensores/Frecuencia_Respiratoria", 0.0);

Firebase.setFloat(firebaseData, path + "/Exámenes/Electrocardiograma", 0.0);
Firebase.setFloat(firebaseData, path + "/Exámenes/Reflejo_Pupilas", 0.0);

Firebase.setString(firebaseData, path + "/Mensajes/Grupo 4", "hola mundo");

```

Figura 37: Anexo 4. Inicialización con Firebase y definición de nodos.

```

//Inicialización ADC

pinMode(Potenciometer, INPUT);
pinMode(pulso_sensor, INPUT);
pinMode(gas_sensor, INPUT);

//config. ADC
analogReadResolution(12); //resolution ADC (9-12)
//analogSetCycles(8);      //nro of cycles per samples (1-255)
//analogSetSamples(1);     //nro of samples in range
//analogSetClockDiv(1);    //divider adc clk
//analogSetAttenuation(); //ADC_0db, ADC_2_5db, ADC_6db, ADC_11db.
//analogSetPinAttenuation(PIN, ATTEN);
//ReadVoltage(pin) //-> respuesta lineal en el rango 0-3V con error < 1%

//0 db = voltaje hasta 1.1V.
//2.5 db = voltaje hasta 1.5V.
//6 db = voltaje hasta 2.2V.
//11db = voltaje hasta 3.9V (limitado por VDD_A). [default]

```

Figura 38: Anexo 5. Inicialización y configuración del ADC.

```

//Loop 2: Transmision wi-fi
void loop() {

    //read ADC, accuracy 7%
    valor_sensado= analogRead(Potenciómetro);
    voltaje_sensado = valor_sensado/4095 *3.3;
    pulso_sensado = analogRead(pulso_sensor);
    gas_sensado = analogRead(gas_sensor);
    breath = -1*gas_sensado;
    enable_data=true;

    //enviar datos
    if (enable_data == true){

        Firebase.setFloat(firebaseData, path + "/Sensores/Potenciómetro", voltaje_sensado);
        Firebase.setInt(firebaseData, path + "/Sensores/Frecuencia_Cardiaca", pulso_sensado);
        Firebase.setInt(firebaseData, path + "/Sensores/Frecuencia_Respiratoria", breath);
        delay(200);

        enable_data = false;
    }
}

```

Figura 39: Anexo 6. Serial Loop de sensado y envío de información a la base de datos.

```

Highcharts.chart('container_1', {
    chart: {
        marginTop: 20,
        type: 'spline',
        animation: Highcharts.svg, // don't animate in old IE
        events: {
            load: function () {
                // set up the updating of the chart each second
                var series = this.series[0];
                // Get a reference to the database service
                var database = firebase.database();
                var Frecuencia_Respiratoria;

                setInterval(function () {
                    database.ref('/Planificación II/Sensores').on("value", function(snap){
                        Frecuencia_Respiratoria = snap.val().Frecuencia_Respiratoria;
                    });

                    var x = (new Date()).getTime(), // current time
                        y = Frecuencia_Respiratoria;
                    series.addPoint([x, y], true, true);
                }, 1500);
            }
        }
},

```

Figura 40: Anexo 7. Código Javascript para lectura de base de datos y asociación al gráfico.

```
//oximeter libraries
#include <Wire.h>
#include <MAX30105.h>
#include <spo2_algorithm.h>
```

Figura 41: Anexo 8. Inclusión de nuevas librerías.

```
//oximeter parameters
MAX30105 particleSensor;

#define MAX_BRIGHTNESS 255

uint32_t irBuffer[100]; //infrared LED sensor data
uint32_t redBuffer[100]; //red LED sensor data

int32_t bufferLength; //data length
int32_t spo2; //SPO2 value
int8_t validSPO2; //indicator to show if the SPO2 calculation is valid
int32_t heartRate; //heart rate value
int8_t validHeartRate; //indicator to show if the heart rate calculation is valid

byte pulseLED = 11; //Must be on PWM pin
byte readLED = 13; //Blinks with each data read

bool first_read =0;
int count_ox =0;

int value_spo2 =0;
int value_bpm = 0;
int sensor_spo2 =0;
int sensor_bpm =0;
```

Figura 42: Anexo 9. Parámetros empleados en el oxímetro.

```

//variables mq135
int sensor_gas = 0;

int contador_resp =0;
int frecuencia_resp= 0;

unsigned long previousMillis=0;
unsigned long intervalos=6000;      //1 segundo
unsigned long ciclos= 15000; //15 segundos
|
bool ref = 0;

```

Figura 43: Anexo 10. Variables sensor de gas

```

//oximeter init
// Initialize sensor
// Initialize sensor
if (!particleSensor.begin(Wire, I2C_SPEED_FAST)) //Use default I2C port, 400kHz speed
{
    Serial.println(F("MAX30105 was not found. Please check wiring/power."));
    while (1);
}

byte ledBrightness = 60; //Options: 0=Off to 255=50mA
byte sampleAverage = 4; //Options: 1, 2, 4, 8, 16, 32
byte ledMode = 2; //Options: 1 = Red only, 2 = Red + IR, 3 = Red + IR + Green
byte sampleRate = 100; //Options: 50, 100, 200, 400, 800, 1000, 1600, 3200
int pulseWidth = 411; //Options: 69, 118, 215, 411
int adcRange = 4096; //Options: 2048, 4096, 8192, 16384

particleSensor.setup(ledBrightness, sampleAverage, ledMode, sampleRate, pulseWidth, adcRange);

```

Figura 44: Anexo 11. Inicialización oxímetro.

```

//Loop 2: Transmision wi-fi
void loop() {

    bufferLength = 100; //buffer length of 100 stores 4 seconds of samples running at 25spS

    if (firstRead == 0)
    {
        //read the first 100 samples, and determine the signal range
        for (byte i = 0 ; i < bufferLength ; i++)
        {
            while (particleSensor.available() == false) //do we have new data?
                particleSensor.check(); //Check the sensor for new data

            redBuffer[i] = particleSensor.getRed();
            irBuffer[i] = particleSensor.getIR();
            particleSensor.nextSample(); //We're finished with this sample so move to next sample

            Serial.print(F("red="));
            Serial.print(redBuffer[i], DEC);
            Serial.print(F(", ir="));
            Serial.println(irBuffer[i], DEC);
        }

        //calculate heart rate and SpO2 after first 100 samples (first 4 seconds of samples)
        maxim_heart_rate_and_oxygen_saturation(irBuffer, bufferLength, redBuffer, &spo2, &validSpO2, &heartRate, &validHeartRate);
        firstRead = 1;
    }
}

```

Figura 45: Anexo 12. Void Loop Parte 1

```

//Continuously taking samples from MAX30102. Heart rate and SpO2 are calculated every 1 second

//dumping the first 25 sets of samples in the memory and shift the last 75 sets of samples to the top
for (byte i = 25; i < 100; i++)
{
    redBuffer[i - 25] = redBuffer[i];
    irBuffer[i - 25] = irBuffer[i];
}

//take 25 sets of samples before calculating the heart rate.
for (byte i = 75; i < 100; i++)
{
    while (particleSensor.available() == false) //do we have new data?
        particleSensor.check(); //Check the sensor for new data

    digitalWrite(readLED, !digitalRead(readLED)); //Blink onboard LED with every data read

    count_ox = count_ox +1;

    redBuffer[i] = particleSensor.getRed();
    irBuffer[i] = particleSensor.getIR();
    particleSensor.nextSample(); //We're finished with this sample so move to next sample
}

```

Figura 46: Anexo 13. Void Loop Parte 2.

```

if (count_ox == 5)
{
    //read ADC, accuracy 7%: Sensado
    valor_sensado= analogRead(Potenciometer);
    voltaje_sensado= valor_sensado/4095 *3.3;
    pulso_sensado = analogRead(pulso_sensor);
    gas_sensado = analogRead(gas_sensor);
    breath = -1*gas_sensado + 2600;
    enable_data=true;

    if (millis() - previousMillis <= ciclos ){

        if (breath > 60 && ref == 0){
            contador_resp = contador_resp +1;
            ref = 1;

        }

        if (breath < -50){
            ref = 0;
        }

    }
}

```

Figura 47: Anexo 14. Void Loop Parte 3.

```

if (millis() - previousMillis > ciclos ){
    Serial.println("breath per 15 seconds: ");
    Serial.println(contador_resp);
    frecuencia_resp = contador_resp * 4;
    Serial.println("frequency respiratory: ");
    Serial.println(frecuencia_resp);

    if (frecuencia_resp > 30){
        frecuencia_resp = 0;
    }

    unsigned long currentMillis= millis();
    previousMillis= currentMillis;
    contador_resp= 0;

}

```

Figura 48: Anexo 15. Void Loop Parte 4.

```

//enviar datos
if (enable_data == true){

    Firebase.setFloat(firebaseData, path + "/Sensores/Potenciómetro", voltaje_sensado);
    Firebase.setInt(firebaseData, path + "/Sensores/Frecuencia_Respiratoria", breath);
    Firebase.setInt(firebaseData, path + "/Resultados/value_rpm", frecuencia_resp);

    if (validspo2 == 1) //lectura valida del oxímetro
    {
        if (irBuffer[i]>5000)
        {
            value_spo2 = spo2;
            sensor_spo2= spo2;
        }

        if (irBuffer[i]<5000) //no finger case
        {
            value_spo2 =0;
            sensor_spo2 =0;
        }

        Firebase.setInt(firebaseData, path + "/Resultados/value_spo2", value_spo2);
        Firebase.setInt(firebaseData, path + "/Sensores/Sat_Oxígeno", sensor_spo2);
    }
}

```

Figura 49: Anexo 16. Void Loop Parte 5.

```

if (validHeartRate == 1) //lectura valida heart rate
{
    if (irBuffer[i]>5000)
    {
        value_bpm = heartRate;
        sensor_bpm = heartRate;
    }

    if (irBuffer[i]<5000) //no finger case
    {
        value_bpm =0;
        sensor_bpm =0;
    }

    Firebase.setInt(firebaseData, path + "/Resultados/value_bpm", value_bpm);
    Firebase.setInt(firebaseData, path + "/Sensores/Frecuencia_Cardiaca", sensor_bpm);
}

enable_data = false;
}

count_ox=0;

```

Figura 50: Anexo 17. Void Loop Parte 6.

```

jQuery(document).ready(function () {
    var $ = jQuery;
    var myRecorder = {
        objects: {
            context: null,
            stream: null,
            recorder: null
        },
        init: function () {
            if (null === myRecorder.objects.context) {
                myRecorder.objects.context = new (
                    window.AudioContext || window.webkitAudioContext
                );
            }
        },
        start: function () {
            var options = {audio: true, video: false};
            navigator.mediaDevices.getUserMedia(options).then(function (stream) {
                myRecorder.objects.stream = stream;
                myRecorder.objects.recorder = new Recorder(
                    myRecorder.objects.context.createMediaStreamSource(stream),
                    {numChannels: 1}
                );
                myRecorder.objects.recorder.record();
            }).catch(function (err) {});
        },
    };

```

Figura 51: Anexo 18. Función para grabar audios parte 1.

```

    stop: function (listObject) {
        if (null !== myRecorder.objects.stream) {
            myRecorder.objects.stream.getAudioTracks()[0].stop();
        }
        if (null !== myRecorder.objects.recorder) {
            myRecorder.objects.recorder.stop();

            // Validate object
            if (null !== listObject
                && 'object' === typeof listObject
                && listObject.length > 0) {
                // Export the WAV file
                myRecorder.objects.recorder.exportWAV(function (blob) {

                    var url = (window.URL || window.webkitURL)
                        .createObjectURL(blob);

                    // Prepare the playback
                    var audioObject = $('<audio controls></audio>')
                        .attr('src', url);

                    // Prepare the download link
                    var downloadObject = $('<a>#9660;</a>')
                        .attr('href', url)
                        .attr('download', new Date().toUTCString() + '.wav');

                    // Wrap everything in a row
                    var holderObject = $('<div class="row"></div>')
                        .append(audioObject)
                        .append(downloadObject);

                    // Append to the list
                    listObject.append(holderObject);
                });
            }
        }
    }
};


```

Figura 52: Anexo 19. Función para grabar audios parte 2.

```

////// CHOICE ///////
var files = [];
document.getElementById("files").addEventListener("change", function(e) {
  files = e.target.files;
  for (let i = 0; i < files.length; i++) {
    console.log(files[i]);
  }
});

////// SEND FILES ///////
document.getElementById("send").addEventListener("click", function() {

  var metadata ={
    contentType: 'audio/wav',
    name: new Date().toUTCString()
  };
  if (files.length != 0) {
    for (let i = 0; i < files.length; i++) {
      var storage = firebase.storage().ref(files[i].name);
      var upload = storage.put(files[i],metadata);
      upload.on("state_changed", function progress(snapshot) {
        var percentage = (snapshot.bytesTransferred / snapshot.totalBytes) * 100;
        document.getElementById("progress").value = percentage;
      },
      function error() {alert("Error uploading file.");},
      function complete() {document.getElementById( "uploading").innerHTML += `${files[i].name} uploaded <br />`}; }
    } else {alert("No file chosen.");}
  });
});

```

Figura 53: Anexo 20. Función para subir archivos al buzón y enviarlos a Firebase Storage.

```

// LIST FILES /////////////////////
document.getElementById("list_files").addEventListener("click", function() {
  var storageRef = firebase.storage().ref();
  storageRef.listAll().then(function(result) {
    result.items.forEach(function(urlFile) {
      var lista = [];

      document.getElementById("response_list").innerHTML += `${urlFile} <br />`;
    }).catch(function(error) {alert("No file chosen.");});
  });
});

///////////////////////////////
// GET URL ///////////////////
document.getElementById("get_url").addEventListener("click", function() {
  var get_url_file = document.getElementById("name").value;
  var audio = document.getElementById("name").value;
  var storage = firebase.storage().ref(get_url_file);
  storage.getDownloadURL().then(function(url) {
    var audio = new Audio(url);
    audio.play();
  }).catch(function(error) {alert("Error encountered.");});
});

```

Figura 54: Anexo 21. Funciones empleadas para leer datos de Firebase Storage.

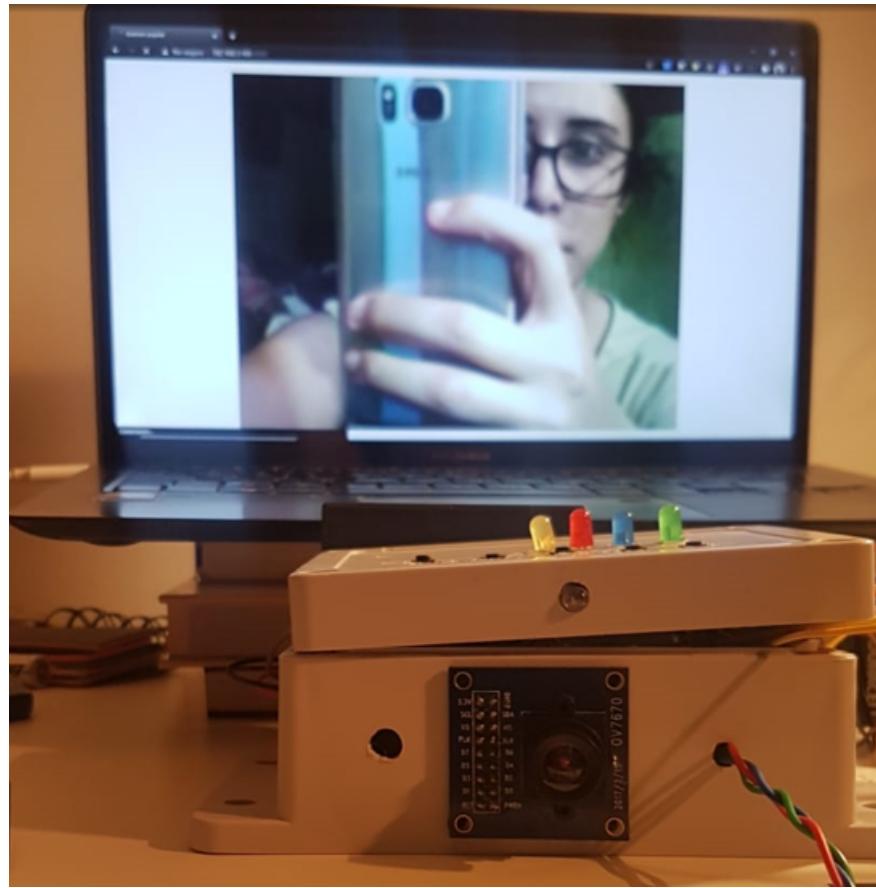


Figura 55: Anexo 22. Streaming con la cámara