



## Monitor de temperatura, ritmo cardiaco y ubicación enviando mensajes de texto a un teléfono celular

### Monitor temperature, heart rate and location by sending text messages to a cell phone

#### **Ricardo Luna Santos**

Tecnológico Nacional de México/Instituto Tecnológico Superior de Huauchinango, Huauchinango, México  
ricardolunasan@gmail.com.mx  
ORCID: 0000-0001-8280-6769

#### **Aldo Hernández Luna**

Tecnológico Nacional de México/Instituto Tecnológico Superior de Huauchinango, Huauchinango, México  
aldo.hdz@huauchinango.tecnm.mx  
ORCID: 0000-0002-7717-5314

#### **Jacinto Torres Jiménez**

Tecnológico Nacional de México/Instituto Tecnológico Superior de Huauchinango, Huauchinango, México  
jacinto.torres@huauchinango.tecnm.mx

#### **Manuel Cruz Luna**

Tecnológico Nacional de México/Instituto Tecnológico Superior de Huauchinango, Huauchinango, México  
manuelcl@huauchinango.tecnm.mx  
ORCID: 0000-0002-0640-8926

doi: <https://doi.org/10.36825/RITI.09.18.007>

Recibido: Enero 21, 2021

Aceptado: Mayo 25, 2021

**Resumen:** En el presente trabajo de investigación se propone el desarrollo de un dispositivo electrónico que permite realizar la lectura de temperatura, ritmo cardiaco y ubicación de una persona y enviar los datos en forma de mensaje de texto a un teléfono celular (*smartphone*) vinculado por medio de la red móvil. El dispositivo de monitoreo que se propone utiliza sensores que nos ayudan a detectar diferentes tipos de señales y que se traducen a lenguaje humano utilizando microcontroladores, electrónica y diversos tipos de programación para el desarrollo de funciones que permitan la obtención de los datos. Como resultado el dispositivo enviará un mensaje de texto solo si las lecturas devuelven un parámetro diferente a los rangos normales del ser humano o cuando el portador del dispositivo presione un botón de ayuda, de esa forma los mensajes de texto llegan al número telefónico del dispositivo móvil vinculado con la información del paciente o familiar.

**Palabras clave:** Monitoreo, Señales, Tiempo Real, Sensores, Red Móvil.

**Abstract:** In this research work, the development of an electronic device that allows reading the temperature, heart rate, and location of a person, and send this data by a text message to a smartphone using a mobile network is proposed. The proposed monitoring device uses sensors that help us detect different types of signals which are translated into human language using microcontrollers, electronics, and various types of programming processes to develop functions to obtain data. As a result, the device will send a text message only if the readings return a parameter different from the normal ranges or when the wearer of the device presses a help button. This will send text messages to the phone number linked to a patient or a patient's relative.

**Keywords:** *Monitoring, Signals, Real-Time, Sensors, Mobile Network.*

## 1. Introducción

El cuidado de la salud es fundamental para que el organismo mantenga una buena calidad de vida. De acuerdo con la Organización Mundial de la Salud (OMS) las enfermedades cardíacas, los infartos, el cáncer, las enfermedades respiratorias y la diabetes, son las principales causas de mortalidad con un 63% de las muertes en el mundo. En 2008, 36 millones de personas murieron de una enfermedad crónica, de las cuales el 50% era de sexo femenino y el 29% era menor de 60 años.

Los signos vitales (SV) son valores que permiten estimar la efectividad de la circulación, de la respiración y de las funciones neurológicas [1]. Para medir estos parámetros existe una variedad de equipos biomédicos que permiten detectar, procesar y desplegar de manera continua cualquier cambio en las condiciones de salud del paciente, los datos pueden ser transmitidos a un computador por medio de puertos de conexiones o vía inalámbrica (Bluetooth) [2]. Dependiendo de su configuración estas unidades pueden medir y mostrar en pantalla las cifras de la presión arterial independiente (PANI), la cantidad de oxígeno transportado por las células rojas de la sangre (SpO<sub>2</sub>) y la temperatura (T).

Actualmente en el mercado están disponible pulseras y relojes inteligentes (Reloj *Smart Watch*) que permiten medir la presión arterial, ritmo cardíaco y hasta generar estadísticas de arritmias, sin embargo, no están diseñados para generar estadísticas en tiempo real de la lectura de signos vitales y no tienen un botón de ayuda que permita enviar una alerta al celular vinculado de otra persona [3].

El dispositivo electrónico que se propone en el presente artículo está dirigido especialmente a las personas con enfermedades crónico-degenerativas, personas de la tercera edad (adultos mayores), infantes o cualquier persona que se preocupe por su salud o la seguridad de sus seres queridos y que necesiten ser monitoreado constantemente para conocer su ubicación y su estado de salud con la finalidad de estar alerta a cualquier síntoma anormal que presenten y de esta forma puedan recibir atención inmediata y oportuna. El dispositivo electrónico tiene las siguientes dos principales características: 1) permitir al portador, presionar un botón de ayuda el cual enviará una notificación por la red móvil que se visualizará en el teléfono celular que se encuentre vinculado y 2) enviar la información del ritmo cardíaco, la temperatura y la ubicación del portador en tiempo real.

**Tabla 1.** Comparación de funciones del dispositivo de monitoreo vs soluciones en el mercado.

Función	Dispositivo electrónico propuesto	Smart Watch	Band	Pulsioxímetro
Pulso	X	X	X	X
Temperatura	X			
Ubicación	X	X	X	
Portable	X	X	X	X
Historial		X	X	
Botón de Ayuda	X	X	X	
Notificaciones al cuidador	X			
Notificaciones al portador		X	X	X

Fuente: Elaboración propia

En la Tabla 1 se presenta un estudio comparativo de las principales funciones del dispositivo electrónico propuesto y de las pulseras y relojes inteligentes existente en el mercado. Como se puede observar la mayoría de las pulseras

y relojes inteligentes no miden la temperatura del portador tampoco tiene la función de enviar notificaciones al cuidador.

## 2. Materiales y métodos

Para la realización del dispositivo electrónico se necesita seguir una serie de pasos que permitirán el desarrollo y diseño de un dispositivo electrónico capaz de leer parámetros relacionados con la temperatura, ritmo cardiaco, ubicación y utilizar la red móvil para enviar notificaciones en forma de mensaje de texto a un *smartphone* o teléfono celular vinculado al dispositivo electrónico.

Para el desarrollo del prototipo se implementó la metodología XP (*Extreme Programming*) con sus cuatro etapas de investigación: 1) análisis, 2) diseño, 3) desarrollo y 4) pruebas. En la Figura 1 se presentan el conjunto de actividades que se deben realizar en cada etapa. La ventaja de utilizar la metodología XP estriba en que el desarrollo del trabajo es ágil y tiene como finalidad entregar valor en períodos cortos de tiempo. La metodología se sustenta en la transparencia, la inspección y la adaptación [4]:

- La transparencia permite dar a conocer toda la información correspondiente al proyecto a las partes interesadas o que integran el grupo de desarrollo.
- La inspección garantiza que las revisiones sean periódicas con la finalidad de detectar y corregir fallos.
- La adaptación tiene que ver con la característica del equipo de trabajo para ajustar tiempos y lograr dichos objetivos.
- 

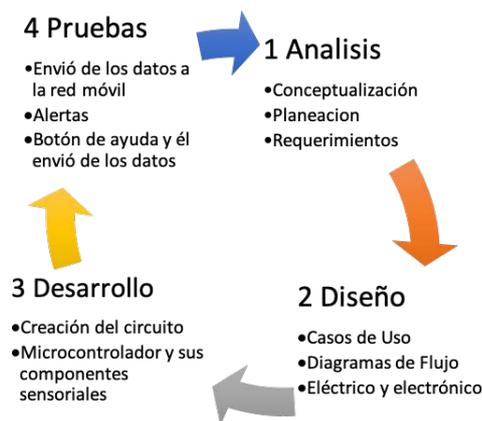


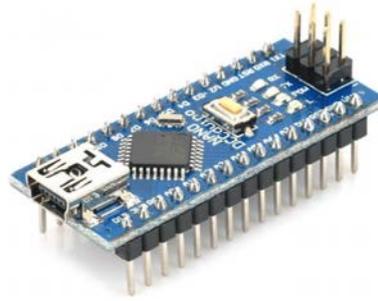
Figura 1. Metodología *Extreme Programming* (o XP).

### 2.1. Análisis

El análisis es el punto de partida para la puesta en marcha de un proyecto y constituye la evaluación de las posibles soluciones para su viabilidad [5]. Los principales rubros del análisis son:

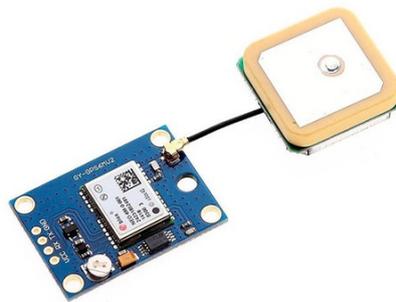
- Conceptualización
- Planeación
- Requerimientos del proyecto

Para el desarrollo del proyecto se utilizaron diferentes tipos de sensores y dispositivos electrónicos los cuales en conjunto con una codificación adecuada conforman un sistema de comunicación embebido que permite la interpretación de señales eléctricas a lenguaje humano por medio del envío y recepción de datos. En la primera etapa se utiliza el dispositivo Arduino, ver Figura 2, debido a que es una plataforma de desarrollo basada en una placa electrónica de hardware libre, el modelo utilizado en el prototipo es Arduino nano que contiene un microcontrolador ATmega328P de bajo costo y versatilidad [6].



**Figura 2.** Placa Arduino Nano.

Con la finalidad de controlar la ubicación del dispositivo se utiliza el módulo GPS6MV2, ver Figura 3, este dispositivo electrónico utiliza el sistema de posicionamiento global (GPS) para determinar la localización exacta de un vehículo, persona, u otro activo, el módulo está basado en el receptor de la marca Ublox modelo NEO 6M, y puede funcionar con un voltaje de alimentación en el rango de 3.0 a 5.0 volts.



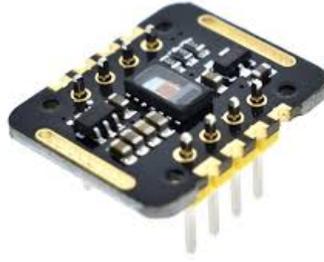
**Figura 3.** Modulo GPS6MV2.

La temperatura corporal es resultado del equilibrio entre la producción y la pérdida de calor del cuerpo, para mantener el control de este parámetro y adquirir una lectura utilizaremos la medición de temperatura de la piel, mediante el sensor MLX90614, ver Figura 4, el cual tiene la función de un termómetro infrarrojo y su principal característica es la medición de temperatura sin contacto.



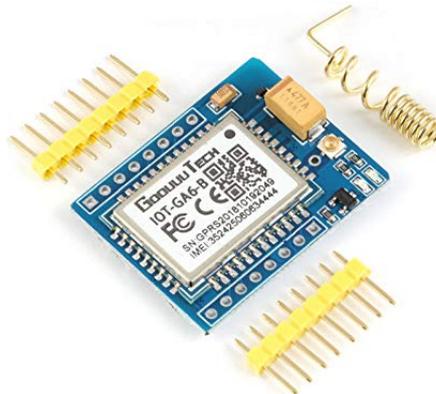
**Figura 4.** Sensor de temperatura infrarrojo MLX90614.

Otro parámetro que tomaremos en cuenta es la frecuencia cardiaca a la cual se identifica como el número de contracciones cardiaca por unidad de tiempo, consiste en el conteo de la cantidad de contracciones cardíacas durante un minuto. El Módulo integrado MAX30102, ver Figura 5, es un detector que mide la frecuencia del pulso haciendo uso de dos Leds: un led rojo (660nm) y un led infrarrojo (920nm), un fotodetector, óptica especializada, filtro de luz ambiental entre 50 y 60Hz, y un conversor ADC delta sigma de 16 bits y de hasta 1000 muestras por segundo en las cuales se mide la cantidad de luz reflejada.



**Figura 5.** Pulsioxímetro MAX30102.

La comunicación para el envío de la información del dispositivo electrónico a un dispositivo móvil se realizará por medio de GPRS, es una tecnología probada y funcional que permite conectividad total, Acceder a Internet, Conectarte con cámaras digitales (capturar y enviar imágenes), Conectarte con reproductores de MP3, Acceder a contenidos informativos y servicios en línea vía WAP, en cualquier momento y lugar. El módulo IoT G A6 GSM GPRS, ver Figura 6, será utilizado para este prototipo y será el encargado de realizar el envío de los datos al dispositivo móvil.

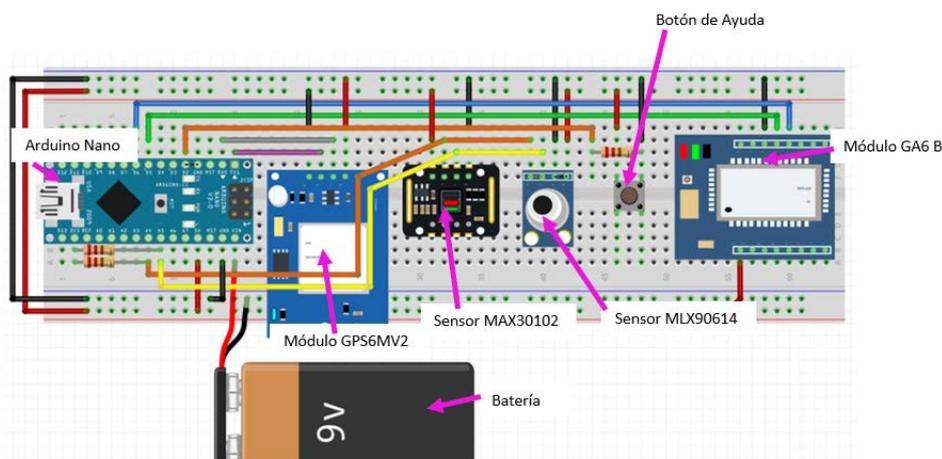


**Figura 6.** Modulo IoT GA6- B GSM/GPRS.

## 2.2. Desarrollo del prototipo

### 2.2.1. Diseño eléctrico y electrónico

Con la ayuda de una placa de pruebas (*protoboard*) se desarrolla la topología electrónica de los sensores y módulos que conforman el prototipo, ver Figura 7, donde la placa de Arduino nano es la encargada de realizar la sincronización a través de código de cada una de las funciones que realizarán los módulos.



**Figura 7.** Diagrama electrónico para envío de ubicación, temperatura, y ritmo cardiaco a través de un mensaje de texto.

Dentro del diseño principal utilizaremos los puertos analógicos, digitales, envío TX y recepción RX de la placa Arduino, como primera parte los puertos TX y RX de la placa van conectados directamente a los puertos TX y RX del Módulo GPS6MV2 tomando en cuenta una conexión cruzada donde el TX de la Placa Arduino se conecta con RX del Módulo y el puerto RX de Arduino se conecta al TX de modulo GPS el módulo cuenta con otros dos pines GND que va al negativo de la batería y VCC que tiene que ir conectado a 5 volts.

El segundo sensor que conectaremos el de pulsioxímetro MAX30102 este módulo cuenta con múltiples pines de los cuales solo utilizaremos el VCC que va conectado directamente a 5 volts, el pin GND que va conectado al negativo de la batería, posteriormente tenemos el puerto SCL que va conectado al pin Analógico 4 de la placa Arduino y el SDA conectado al puerto Analógico 5 de la placa Arduino. Para este sensor se realiza una conexión de tipo I2C que es un protocolo síncrono. I2C usa solo 2 cables, esto puede ocasionar problemas con la comunicación a través del mismo medio con otros sensores por esta razón se utiliza resistencias de pull-up esto permitirá poner el bus a nivel alto, y los dispositivos envían niveles bajos. Si quieren enviar un nivel alto simplemente lo comunican al bus y será reconocido ya que cada sensor tiene una dirección única que permitirá trabajar las señales de múltiples sensores por medio de dos cables.

El tercer sensor que también utiliza una comunicación I2C es el sensor infrarrojo de temperatura MLX90614 se conecta de la siguiente manera pin VCC a 5 volts, pin GND al negativo de la batería, pin SCL al pin Analógico 4 de Arduino y SDA al pin analógico 5 de la placa Arduino como puedes ver este sensor comparte los puertos de comunicación con otro, esto es posible gracias al protocolo I2C.

Como cuarto paso conectaremos el módulo que permitirá el envío de las lecturas de los sensores procesadas por el microcontrolador en forma de mensaje de texto a un dispositivo móvil, estamos hablando del módulo GA6-B GSM/GPRS este hace uso de un chip para el acceso a la red móvil y está conectado a la placa Arduino por medio de una conexión serial que utiliza el puerto URX y UTX del módulo conectado a los puertos digitales 5 y 6 de la placa Arduino, un puerto GND conectado al negativo de la batería y un puerto VCC que se conecta directamente a 5 volts, ver Figura 8.

Por último, se integra la conexión del botón de ayuda que va de un extremo al GND de la batería, seguido de una resistencia la cual conecta al otro extremo al botón y directamente al pin digital 2 de la placa Arduino. La conexión de estos módulos y sensores crean un dispositivo electrónico capaz de realizar lecturas de ubicación, temperatura, frecuencia cardiaca, procesar estas señales con el microcontrolador de la placa Arduino y enviarlas por la red móvil en forma de mensaje de texto a un teléfono celular o smartphone de forma automática cuando se detectan variaciones fuera de lo común en las lecturas de los signos vitales o en su defecto cuando el portador del dispositivo oprime el botón de ayuda.

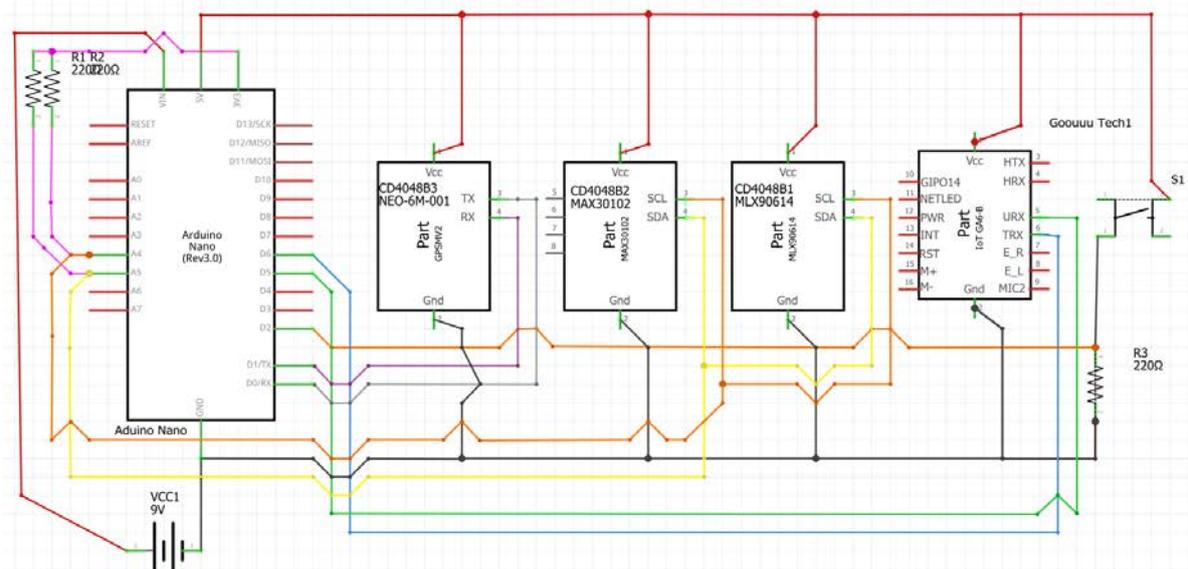


Figura 8. Diagrama eléctrico del prototipo.

### 2.2.2. Programación

La programación permite ayudar a elegir una decisión de forma correcta, gracias a este proceso podemos desarrollar y establecer un conjunto de actividades en un tiempo determinado, garantizar la ejecución de las tareas y alcanzar los objetivos establecidos. La programación se deriva de un proceso continuo en el que interfieren varios factores de organización, planeación, desarrollo y ejecución [7].

En el mundo actual la automatización de procesos demanda de la programación de funciones, estas líneas de código permiten la comunicación entre el microcontrolador, módulos o sensores, con la finalidad de interpretar las lecturas analógicas o digitales, tomar decisiones, procesar la información, mostrarla o enviarla en un formato entendible para el ser humano, realizar conversiones, operaciones, análisis y tratamiento de los datos con la finalidad de establecer un intercambio de información, esto es posible gracias a la codificación.

Para el desarrollo de este proyecto utilizaremos cuatro funciones principales que corresponden a cada módulo, serán programadas en la placa de desarrollo Arduino y se describe su funcionamiento a continuación. Para mejorar la comprensión del código en nuestro microcontrolador compartimos el repositorio público almacenado en la nube para su consulta <https://github.com/rlunas/Ubics>

- La placa Arduino Nano es el motor principal del funcionamiento de nuestro dispositivo electrónico, es el encargado de realizar el procesamiento, inicializar las funciones, realizar la toma de decisiones y comunicar a los módulos o sensores con el procesador central para preparar la información y enviarla a su destino. Para iniciar la codificación en la placa se deben importar las librerías, ver Figura 9, que permite utilizar y gestionar todas las características de los módulos como el de comunicación GSM/GPRS, gestión a la red GPS, monitoreo de señales del sensor de temperatura y pulso cardíaco.

```
#include <SoftwareSerial.h>//incluimos la libreria SoftwareSerial
#include <TinyGPS.h>//incluimos la libreria TinyGPS para el modulo GSMV2
#include <Wire.h> //Libreria que permite comunicarse con dispositivos por bus I2C
#include <MAX30105.h> //Libreria que permite obtener las lecturas de lu reflejada
para calcular la frecuencia cardiaca
#include <spo2_algorithm.h>
#include <heartRate.h>
#include <Adafruit_MLX90614.h>//Libreria que permite utilizar el sensor óptico
```

Figura 9. Importar librerías y declarar variables del sistema.

- El *setup*, ver Figura 10, es la primera función en ejecutarse dentro de un programa en Arduino. Es, básicamente, donde se desarrollan las funciones que llevará a cabo el microcontrolador. Aquí es donde establecemos algunos criterios que requieren una ejecución única, en la función *setup* declaramos el pin 2 digital de Arduino en modo entrada ya que será el responsable de captar el evento del botón de ayuda de esta forma saber si esta pulsado o no, posteriormente establecemos la comunicación con la interfaz serial del Arduino, la conexión serial del módulo GA6 -B GSM/GPRS.
- La función *void loop* es una función que se repite constantemente en forma de un ciclo infinito, dentro de esta función se manda a traer y ejecutar la función ubicación la cual está ligada a otras funciones que permiten obtener lecturas de nuestro módulo y sensores.

```
void setup()
{
  pinMode(boton,INPUT);
  Wire.begin();
  Serial.begin(9600);//Iniciamos el puerto serie
  serialmovil.begin(9600);//Iniciamos el puerto serie del modulo GA6 B GSM/GPRS
  while(!Serial);
}
void loop()
{
  ubicacion();
}
```

Figura 10. Programación de la función *setup* y la función *loop*.

- La función ubicación correspondiente al módulo GPSTMV2 utiliza la librería TinyGPS.h que es fácil de usar para extraer la posición, fecha, hora, altitud, velocidad y rumbo de los dispositivos, se encuentra programada tomando una serie de pasos, ver Figura 11, en la primera parte nombramos a la función como ubicación (). Posteriormente verificamos que la conexión serial esta activa o lista para la comunicación, en caso de que si este disponible realizamos una lectura la cual almacenaremos en la variable c y con la ayuda de la librería para el módulo GPS obtenemos la lectura de latitud y longitud [8] la almacenamos en las variables y damos un tiempo de 500 milisegundos, ejecutamos la función temperatura, posteriormente pasando otros 500 milisegundos, ejecutamos la función frecuencia e inicializamos la variable x en 0 esperamos 500 milisegundos y ejecutamos la función mensaje para el envío de los datos por la red GSM/GPRS.

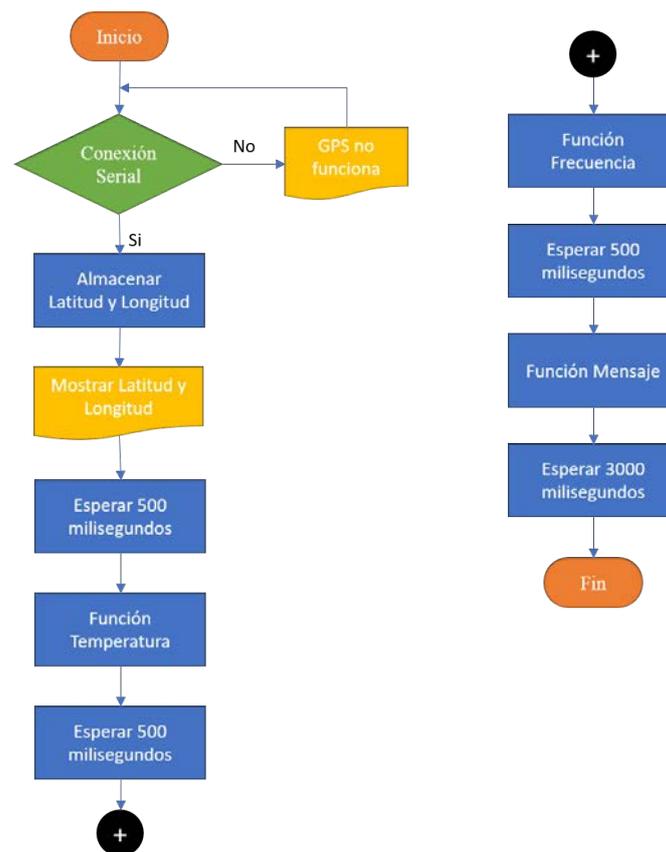


Figura 11. Diagrama de flujo para obtener ubicación.

- Para codificar la estructura que calcula la temperatura con el módulo MLX90614 es necesario crear una función llamada temperatura, ver Figura 12, la cual utiliza la librería *Wire.h* para establecer la comunicación I2C y *Adafruit\_MLX90614.h* para obtener las lecturas se utiliza una variable de tipo flotante llamada *temp* y se inicializa en 0, iniciamos la conexión del dispositivo con *mlx.begin()* la cual se encarga de mostrar la comunicación de las lecturas eléctricas del dispositivo y con el comando *readObjectTempC* obtenemos las lecturas en grados centígrados, posteriormente este valor se asigna a la variable *temp* y está lista para ser mostrada al usuario.



**Figura 12.** Diagrama de flujo función temperatura.

- Para obtener datos que corresponden al sensor de pulso cardiaco MAX30102 se utiliza la librería MAX3010x de *SparkFun* que permite realizar el cálculo de la luz reflejada y analizar los parámetros para calcular el ritmo cardiaco, este sensor igual utiliza la comunicación I2C para la lectura y recepción de señales eléctricas. Iniciamos la codificación de este sensor verificando la conectividad del puerto I2C, si existe una lectura diferente a la señal del sensor se manda el mensaje que indica cuando no está funcionando el sensor de lo contrario si se detecta una señal se pide al usuario que coloque su dedo para iniciar la lectura, se agrega una condición para que el botón de ayuda sea capaz de enviar los datos de ubicación y temperatura aun cuando no se detecte pulso cardiaco, ver Figura 13.



**Figura 13.** Detectar ritmo cardiaco y verificar status del botón de ayuda.

Continuando con la codificación del módulo en el proceso realizar lectura de pulso cardiaco, generamos un ciclo de 1500 repeticiones en las cuales tomaremos lecturas de la luz led IR, con la función obtenemos una lectura y la almacenamos en una variable de tipo long llamada irValue, verificamos si la lectura esta arriba de 50000 de esta forma garantizamos que se está realizando la lectura de la luz reflejada de lo contrario enviara un mensaje recomendando colocar el dedo o la muñeca, una vez detectada la señal asignando un valor verdadero y posteriormente se calcula el tiempo en milisegundos de una lectura a otra, este parámetro se almacena en la variable delta, posteriormente se toman la variable delta y se divide entre 1000.0 que son las muestras que se realizan por segundo, el resultado de este valor se divide entre 60 que es el tiempo de lectura y da como resultado las pulsaciones por minuto [9] ver Figura 14.

Una vez calculado el parámetro BPM se realiza una comparación para saber si el valor es menor que 255 y mayor que 20 si se cumple la condición almacenar el valor en un arreglo y posteriormente calcula el promedio de la lectura, una vez calculados los valores se imprimen en terminal serial cuanto se ejecuta cada lectura se activa una función correspondiente al botón de ayuda el cual al ser pulsado detectara la pulsación, tomara la temperatura y enviara los parámetros pausando el proceso de la toma de ritmo cardiaco dando prioridad al envío por del mensaje.

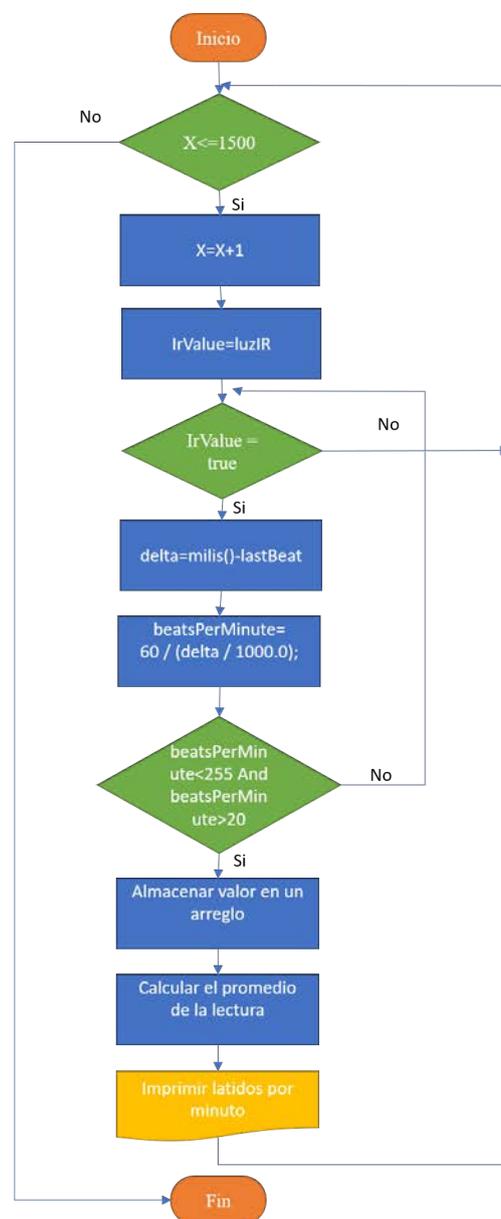


Figura 14. Calcular el ritmo cardiaco.

Una vez que se han desarrollado las codificaciones de todos los módulos y sensores pasamos al envío de los mensajes de texto por medio del módulo GA6 -B GSM/GPRS, ver Figura 15, este envío depende directamente de la interfaz serial llamada serial móvil, ejecutamos los comandos AT+CMGF=1 para configurar un mensaje de texto, para saber el destino del mensaje de texto utilizamos el comando AT+CMGS=\”teléfono al que se envía la información\” [10], posteriormente comparamos la temperatura si un parámetro (temperatura, ritmo cardiaco) este fuera de los rangos normales el mensaje es enviado automáticamente de lo contrario podemos enviar los cuándo se oprima el botón de ayuda en el dispositivo electrónico.

```
void mensaje(){
  serialmovil.println("AT+CMGF=1"); // Configuring modo TEXTO
  serialmovil.println("AT+CMGS=\"+527641114713\"");//cambiar codigo de pais +52 colocar numero a 10 d
  igitos del telefono al que llegaran los sms
  if(temp>37.2 or beatAvg>20 and beatAvg<50 or beatAvg>120){
    serialmovil.print("Temperatura o Frecuencia Fuera de Rango\n"); //COnenido del texto
    serialmovil.println("Ubicacion: https://maps.google.com/?q=");
    serialmovil.print(latitud,5);
    serialmovil.print(", ");
    serialmovil.print(longitud,5);
    serialmovil.println("\nLatidos: ");
    serialmovil.print(beatAvg);
    serialmovil.println("\nTemperatura: ");
    serialmovil.print(temp);
    serialmovil.write((char)26);
    delay(2000);
  }else if(envio==1){
    serialmovil.println("Temperatura y Frecuencia Cardiaca Normal\n"); //COnenido del texto
    serialmovil.println("Ubicacion: https://maps.google.com/?q=");
    serialmovil.print(latitud,5);
    serialmovil.print(", ");
    serialmovil.print(longitud,5);
    serialmovil.println("\nLatidos: ");
    serialmovil.print(beatAvg);
    serialmovil.println("\nTemperatura: ");
    serialmovil.print(temp);
    serialmovil.write((char)26);
    delay(2000);
  }else{
    Serial.println("Ubicacion: ");
    Serial.print(latitud,5);
    Serial.print(",");
    Serial.print(longitud,5);
    Serial.println("\nLatidos: ");
    Serial.print(beatAvg);
    Serial.println("\nTemperatura: ");
    Serial.println(temp);
  }
}
```

Figura 15. Código para el envío del mensaje de texto de forma automática o manual.

### 3. Resultados

Para validar el correcto funcionamiento del prototipo se realizan las pruebas para evaluar el funcionamiento de cada uno de las secciones que componen el prototipo. Como primera parte al iniciar el programa realizamos una conexión serial a la computadora para visualizar las lecturas de los sensores, ver Figura 16, el programa al iniciar nos proporciona los datos de latitud y longitud que permitirán obtener la ubicación del dispositivo, posteriormente se realizan 1500 lecturas de IR la cual tarda alrededor de un minuto, una vez finalizada se ejecuta la función de temperatura y se muestran los parámetros de localización, frecuencia cardiaca y temperatura.

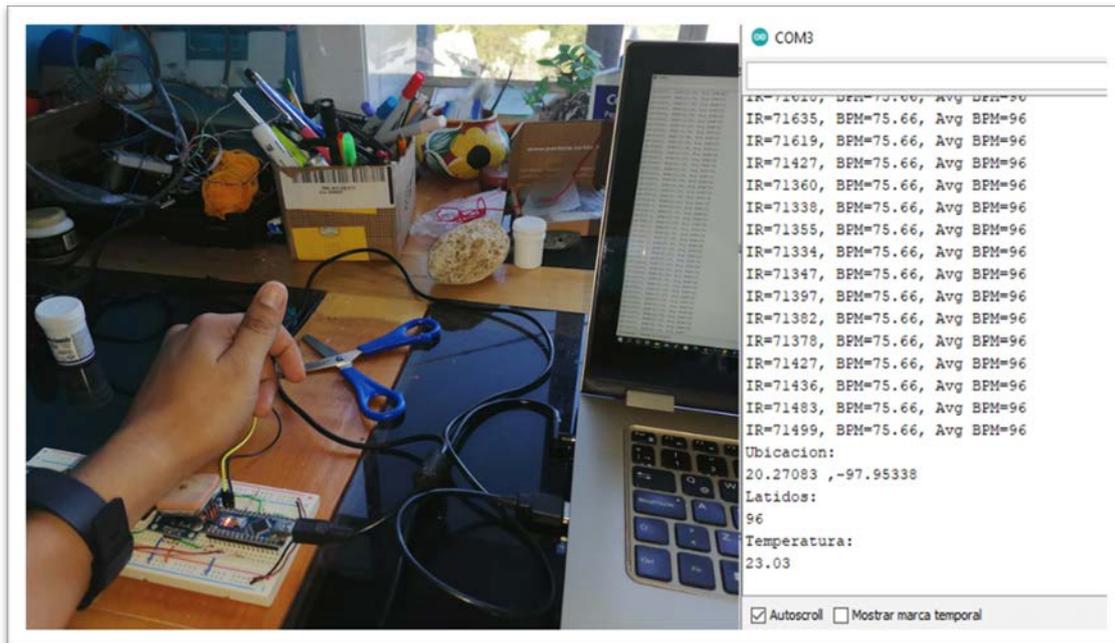


Figura 16. Lectura de los sensores en conexión serial.

La segunda prueba se realiza al presionar el botón de ayuda en el dispositivo y se envían las lecturas al dispositivo móvil indicando si los parámetros están dentro del rango normal o fuera de rango, ver Figura 17.

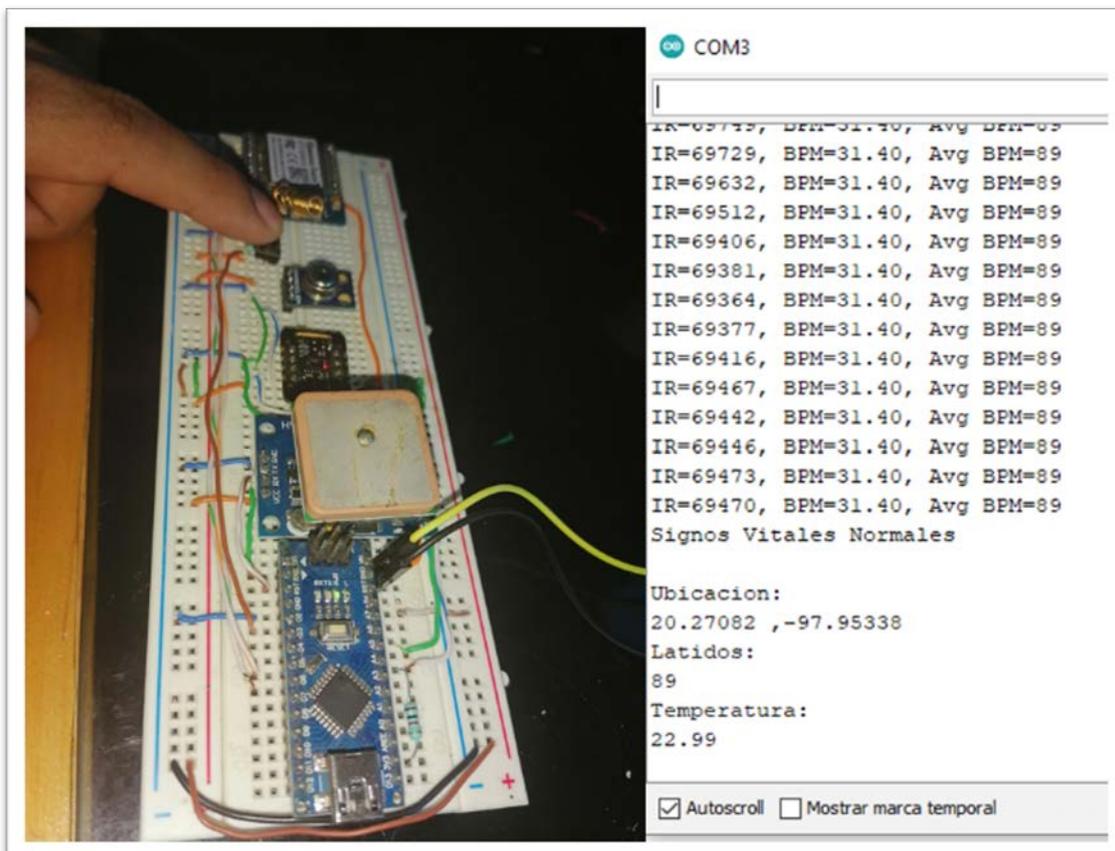


Figura 17. Lecturas obtenidas presionando el botón de ayuda en el dispositivo.

La siguiente prueba realizada al dispositivo se utilizó un cautín y se acercó al sensor de temperatura, ver Figura 18, con la finalidad de sobrepasar el rango normal de los parámetros leídos y automáticamente al detectar esta variación se muestra y envía el mensaje de signos vitales fuera de rango al dispositivo móvil vinculado.

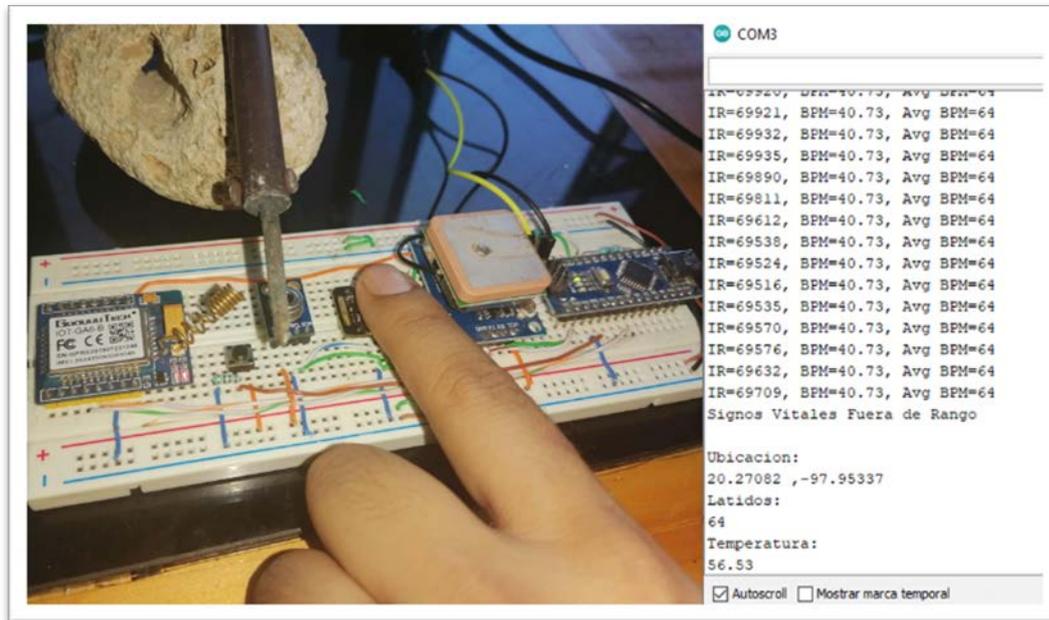


Figura 18. Lecturas fuera de rango obtenidos por la variación de los signos vitales.

Una vez finalizadas las pruebas se mostraron las notificaciones, ver Figura19, en estas imágenes se muestra el mensaje de texto enviado por el dispositivo electrónico vinculado al número telefónico registrado en la función de mensaje del código.

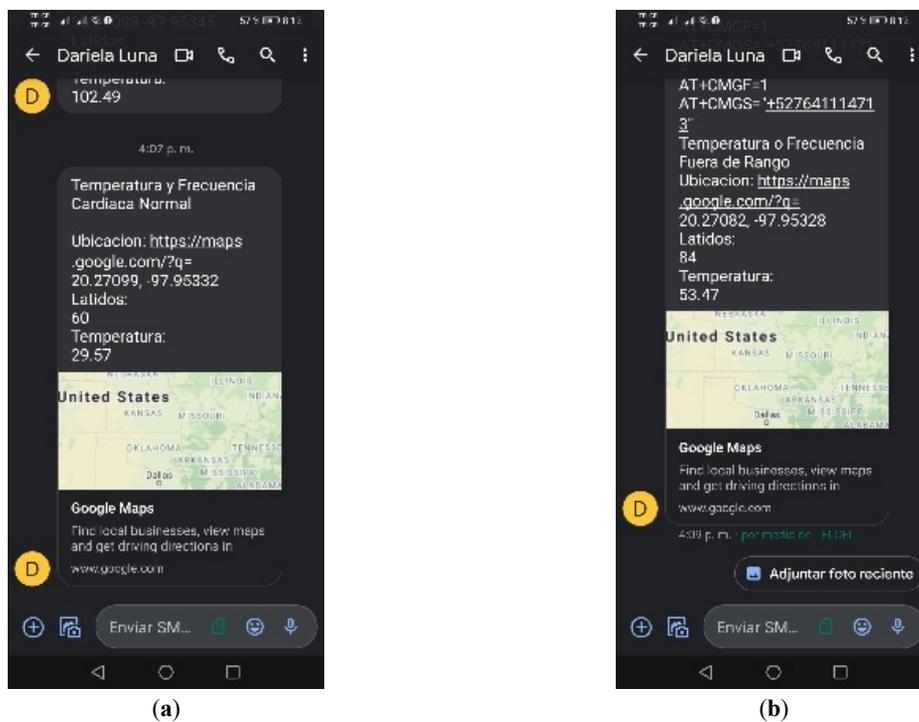


Figura 19. Notificaciones del mensaje en el dispositivo móvil: (a) Notificaciones al presionar el botón de ayuda; (b) notificación por incremento de temperatura o variación de frecuencia cardiaca.

#### 4. Conclusiones

En el presente artículo se presentó el desarrollo de un dispositivo electrónico que permite medir la temperatura y ritmo cardiaco del portador, así como localizar su ubicación. El dispositivo constituye una herramienta útil para las personas encargadas de monitorear la ubicación y el estado de salud de personas que padecen alguna morbilidad crónica, o de adultos o infantes que necesiten ser vigilados para que a cualquier situación anormal puedan recibir atención inmediata y oportuna. Además, el dispositivo propuesto tiene la característica de permitir al portador, presionar un botón de ayuda el cual enviará una notificación en forma de texto que se visualizará en el teléfono celular que se encuentre vinculado.

A las secciones que componen el dispositivo electrónico se le realizaron diversas pruebas para evaluar su correcto funcionamiento logrando resultados prometedores.

#### 5. Referencias

- [1] Villegas González, J., Villegas Arenas, O. A., Villegas González, V. (2012). Semiología de los signos vitales: Una mirada novedosa a un problema vigente. *Archivos de Medicina (Col)*, 12 (2), 221-240.
- [2] ECRI Institute. (2016). *Comparación de monitores de signos vitales*. Recuperado de: <http://www.elhospital.com/temas/Monitores-de-signos-vitales-Parte-1+114421?pagina=1>
- [3] Hon, D. (2019). *Relojes inteligentes: de dispositivos inútiles a guardianes de la salud*. Recuperado de: <https://www.technologyreview.es/s/10982/relojes-inteligentes-de-dispositivos-inutiles-guardianes-de-la-salud>
- [4] Beck, K., Andres, C. (2004). *Extreme Programming Explained: Embrace Change* (2da Ed.). Boston: Addison-Wesley Professional.
- [5] Sommerville, I. (2005). *Ingeniería del software* (7ma Ed.). Madrid: Pearson Educación.
- [6] Arduino (2020). *Arduino Products*. Recuperado de: <https://www.arduino.cc/en/Main/Products>
- [7] Rosales, M. (2020). *Concepto de Programación*. Recuperado de: <https://www.zonaeconomica.com/concepto-programacion>.
- [8] Pérez del Águila, P. L. (2017). *Diseño de un sistema de rastreo embebido en vehículo terrestre, enlazado a servicios en la nube, utilizando el módulo GPS6MV2* (Doctoral dissertation). Universidad de San Carlos de Guatemala.
- [9] González Romero, F. (2019). *Diseño e implementación de un pulsioxímetro reflexivo y estudio de su funcionamiento en diferentes zonas del cuerpo* (Trabajo de Fin de Grado). Universidad de Sevilla, Sevilla.
- [10] Cumbal Simba, R., Buestán, J. C., Domínguez, J. C. (2021). Implementación de una red IoT con GPRS para monitorear los parámetros en un vehículo en tiempo real. *Revista de Investigación en Tecnologías de la Información (RITI)*, 9 (17), 66-76. doi: <https://doi.org/10.36825/RITI.09.17.007>