



Código: PVT-CAyT-01-2020

**“SISTEMA INTELIGENTE DE MONITOREO Y  
ALERTA BIOMÉTRICA (SIMAB) PARA LA  
DETECCIÓN TEMPRANA DE SÍNTOMAS  
ASOCIADOS A COVID 19 Y OTRAS  
ENFERMEDADES EN ADULTOS MAYORES”**

Director: GIUFFRIDA, Pedro Mario

Integrantes: COLL, Pablo Enrique; MARTINEZ, Felipe Raúl;  
ARRUA, Lucas Matías; DI VICENZO, Mariano Nicolás.

Año: 2024



## **Informe Final de Proyectos de Vinculación Tecnológica**

**Universidad Nacional de Moreno**

## Contenido

Identificación del proyecto .....	3
PARTE I - Informe de resultados para el repositorio digital institucional de acceso abierto de la UNM – Ley N° 26.899 .....	4
1. Antecedentes y objetivos.....	4
2. Demandante y sector productivo .....	6
3. Actividades realizadas .....	7
4. Resultados y productos .....	21
5. Impacto.....	21

## Identificación del proyecto

Código:	PVT-CAyT-01-2020
Tipo de proyecto <sup>1</sup> :	PVT
Nombre completo del proyecto:	Sistema Inteligente de Monitoreo y Alerta Biométrica (SIMAB) para la detección temprana de síntomas asociados a COVID 19 y otras enfermedades en adultos mayores
Director/a:	Esp. Ing. Pedro M. Giuffrida
Lineamiento prioritario <sup>2</sup> :	ELECTRÓNICA Y DESARROLLO TECNOLÓGICO
Sub-línea prioritaria	Electrónica aplicada
Organismo financiador:	SPU
Convocatoria:	VT para el desarrollo Local II
Contraparte:	UNM
Fecha de inicio:	17/07/2020
Fecha de finalización:	16/07/2021
Unidad de localización: Departamento/centro/ Programa	Departamento de Ciencias Aplicadas y Tecnología
Resumen:	<p>Actualmente existen diferentes dispositivos para monitorear datos biométricos donde la información capturada es gestionada por el usuario en cuanto a sus alertas y acciones a tomar.</p> <p>Dado que la utilización de estos datos son importantes para tomar decisiones de índole médico, es interesante el desarrollo de un sistema que los almacene en forma individual y los gestione de forma tal de dar una alerta a un Profesional Médico referido de</p>

<sup>1</sup> Tipos de Actividades de Vinculación y Transferencia Tecnológica (AVTT):

PVT: Proyectos de Vinculación y Transferencia

SAT: Servicios a terceros

SE: Servicios Estandarizados

<sup>2</sup> Según LINEAMIENTOS ESTRATÉGICOS DE INVESTIGACIÓN CIENTÍFICA Y DESARROLLO TECNOLÓGICO 2022-2027 (Aprobado por Resolución UNM-CS N° 848/21)

[http://www.unm.edu.ar/files/Lineamientos\\_Estrategicos\\_de\\_Investigacion\\_Cientifica\\_y\\_Desarrollo\\_Tecnologico\\_UNM\\_2022\\_2027\\_Oct\\_2021.pdf](http://www.unm.edu.ar/files/Lineamientos_Estrategicos_de_Investigacion_Cientifica_y_Desarrollo_Tecnologico_UNM_2022_2027_Oct_2021.pdf)

	<p>desvíos en los mismos o bien que el mismo sistema tome la “iniciativa” de alertar a un tercero a definir por el usuario.</p> <p>EL SIMAB se compone de un sistema básico basado en sensores tipo pulsera comunes en el mercado que enviara los datos biométricos y de geolocalización al sistema centralizado de toma de decisión desde un teléfono inteligente.</p> <p>Un sistema avanzado particularmente útil para personas con síntomas asociados a COVID 19 y otras enfermedades en adultos mayores y/o con discapacidades que sean autónomos y activos donde sea necesario un monitoreo constante no invasivo y permita darle una seguridad adicional en su desarrollo diario previniéndolo de tener accidentes de salud, desorientaciones, etc., basado en una pulsera autónoma.</p>
Palabras claves:	Sensores, transmisión, inteligencia artificial, analítica de datos.

## **PARTE I - Informe de resultados para el repositorio digital institucional de acceso abierto de la UNM – Ley N° 26.899**

### ***1. Antecedentes y objetivos***

El presente Proyecto se basa en el PICyDT 2019 “Sistema Inteligente de Monitoreo y Alerta Biométrica (SIMAB)”

#### **Introducción:**

Actualmente existen diferentes dispositivos para monitorear datos biométricos donde la información capturada es gestionada por el usuario en cuanto a sus alertas y acciones a tomar.

Dado que la utilización de estos datos es importante para tomar decisiones de índole médico, es interesante el desarrollo de un sistema que los almacene en forma individual y los gestione de forma tal de dar una alerta a un Profesional Médico referido de desvíos en los mismos o bien que el mismo sistema tome la “iniciativa” de alertar a un tercero a definir por el usuario.

El proyecto SIMAB está pensado como un dispositivo portable no invasivo capaz de tomar mediciones de alta precisión, y transmitir las en tiempo real a una base de datos para su análisis, permitiendo monitorear a pacientes con síntomas asociados a COVID 19 y otras enfermedades

de adultos mayores que necesiten un control ambulatorio permanente de determinadas variables por problemas de salud.

El dispositivo releva indicadores biométricos, tales como oxigenación en sangre, temperatura y pulsaciones del paciente que lo lleva consigo. Junto con estos indicadores biométricos se relevan simultáneamente la geo-localización y la posición relativa del sujeto en los ejes cartesianos (x,y,z).

Los datos relevados son transmitidos inalámbricamente en forma de paquetes en tiempo real a una base de datos que procesa estas variables y actúa en consecuencia a través de alarmas al sistema medico elegido y además guarda un registro completo de la actividad del paciente para su posterior análisis.

Esta información es relevada y revisada por un operador médico que tomara las acciones del caso según los datos capturados y transmitidos por el dispositivo hacia la base de datos.

Una vez colectados y almacenados los datos un motor de inteligencia artificial analiza y compara la información actual con las almacenadas para cada individuo y se entrena a través de una red neuronal pudiendo tomar acciones directas o sugerirlas para que el operador médico las valide.

EL SIMAB se compone de las siguientes etapas:

- Sensores de monitoreo de signos vitales: presión arterial, oxígeno en sangre, pulsaciones Sensores de posición relativa de la persona conectada a un módulo GPS.
- Sistema de gestión y control de los sensores. Sistemas de comunicaciones y transferencia de datos. Motor de inteligencia artificial.
- Red neuronal de aprendizaje.

### **Objetivos:**

Desarrollar una aplicación basada en inteligencia artificial (IA) y sistema de aprendizaje basado en red neuronal que tome los datos biométricos obtenidos desde la red de sensores del usuario tomando decisiones respecto de los mismos.

Desarrollar un dispositivo portátil de precisión para el usuario que obtenga información mediante sensores biométricos de: presión sanguínea, oxígeno en sangre, pulsaciones, caídas (acelerómetro).

Definir el sistema de comunicación para enviar los datos de los sensores al servidor SIMAB para su procesamiento y acción geo posicionando los mismos.

Coadyuvar a la mejora de la calidad de vida de las personas a través de la tecnología, dándole a este la función de mejora social. Crear conciencia de Investigación y desarrollo en los y las Estudiantes, espíritu de colaboración y equipo como así también una visión de futuro hacia la Ciencia y Tecnología y la importancia que esta tiene en la comunidad.

Lograr un desarrollo que implique un a transversalidad de conocimientos entre disciplinas diferentes articulando adecuadamente la interacción entre las misma sinergizando el logro tecnológico.

---

## ***2. Demandante y sector productivo***

---

Monitorear a pacientes con síntomas asociados a COVID 19 y otras enfermedades de adultos mayores que necesiten un control ambulatorio permanente de determinadas variables de salud es de suma utilidad.

El cuidado de la salud en personas de edad avanzadas o con capacidades diferentes puede, en algunos casos, limitar las actividades y libertades de la misma haciendo que su calidad de vida disminuya o que se sienta sin libertad.

La atención temprana de variación de indicadores biométricos dando una alerta adecuada a un centro de salud, su médico de cabecera, un sistema de alerta comunitario (SAME), algún pariente, etc. puede contribuir a prevenir un mal mayor evitando complicaciones de salud.

Es común que algunas personas con afectaciones específicas se encuentren desorientadas o perdidas, en estos tipos de problemas se opta por limitar el movimiento de los mismos; el SIMAB permitirá darle una mayor libertad de movimiento permitiendo que, en caso de desorientación y a través de un botón de pánico se localice a la persona y obtenga la adecuada asistencia.

Actualmente existe en el mercado diferentes sistemas de adquisición de datos biométricos que son locales siendo su información consultada exclusivamente por el individuo usuario de manera tal que, salvo que el mismo este familiarizado con la implicancia que esos datos puedan tener para su salud, se corre el riesgo de pasar por alto datos que pueden ser riesgosos. Además los datos se obtienen de manera instantánea si tener ninguna correlación con datos anteriores, es

decir, no existe ninguna comparativa de los mismos respecto de su evolución temporal careciendo de una recomendación adecuada respecto de la actividad que desarrolla y como se puede afectar su salud.

El SIMAB tendrá la capacidad necesaria y suficiente para interpretar los datos actuales, compararlos con los anteriores y analizar su evolución dando una alerta temprana al usuario o a la institución/personas que el mismo defina tomando las acciones correctivas adecuadas.

Esta última tarea la desarrollará la parte de IA del SIMAB que residirá en servidores adecuados para gestionar los datos con los algoritmos adecuados para la toma de decisión y de contacto.

Así, detectada una anomalía, podrá intervenir un profesional entrenado en salud para tomar las acciones pertinentes evitando agravamientos o localizando a la persona y organizar su búsqueda y rescate de ser necesario.

Respecto del dispositivo de censado, el mismo puede ser una pulsera comercial o utilizar el sistema de sensores a desarrollar que enviara por la red móvil los datos de los diferentes sensores al SIMAB central. Cabe aclarar que, en este último caso, el dispositivo a desarrollar se conecta a un sistema de transmisión móvil celular autónomo para la transferencia de los datos al sistema IA.

---

### ***3. Actividades realizadas***

#### **a) Elección de componentes:**

---

El prototipo del sistema se realizó con una placa de desarrollo que contiene un microcontrolador ESP32. A este microcontrolador se le conectaron los distintos módulos con sus sensores para relevar las variables mencionadas.

El módulo ESP32 tiene conectividad inalámbrica Wi-Fi y Bluetooth Low Energy (BLE). En su interior alberga temporizadores y conversores para procesar señales analógicas y/o digitales. El corazón del módulo ESP32 está formado por un microcontrolador Tensilica Xtensa LX6 de 32 bits que incluye dos núcleos de 240 MHz. [1] [2]

#### **Sensor de Oxígeno en sangre y pulsaciones:**

Para la Oximetría de pulso se eligió el sensor MAX30100. [3]

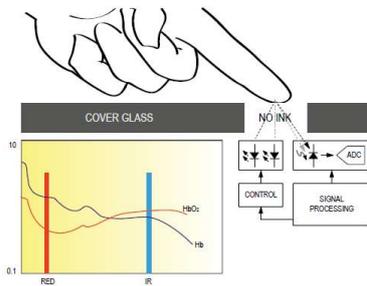


Figura 1: Diagrama en bloques del módulo.

“La oximetría de pulso transcutánea estima la saturación de oxígeno (SpO<sub>2</sub>) de la sangre capilar sobre la base de la absorción de la luz proveniente de diodos emisores de luz colocados en una pinza de dedo o en una sonda con una cinta adhesiva. Las estimaciones suelen ser muy precisas y se correlacionan con una variación menor del 5% de la saturación de oxígeno arterial (SaO<sub>2</sub>) medida.” [4]

La oximetría se basa en la ley de Beer, que establece que la concentración de una sustancia desconocida, disuelta en otra, se puede determinar mediante absorción de luz. Los sensores para oximetría se basan en esta ley cuando detectan la luz que se hace pasar a través de un dedo (Figura 2). La mayor parte de la luz es absorbida por el tejido conectivo, piel, huesos y sangre venosa en forma constante. Sin embargo con cada latido del corazón, hay un pequeño aumento en la sangre arterial, lo que resulta en un aumento en la absorción de luz. [5]

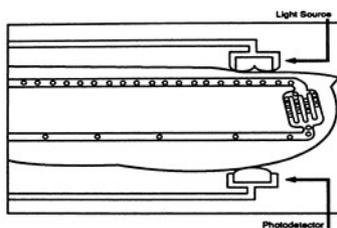


Figura 2: Sonda de oximetría de pulso colocada a través del lecho vascular de un dedo.

“La sonda se compone de una fuente de luz y un sensor (fotodetector). La fuente de luz consta de dos LED que emiten luz en longitudes de onda conocidas, generalmente 660 nm (rojo) y 940 nm (infrarrojos). Se utilizan estas particulares longitudes de onda porque las características de absorción de oxihemoglobina y hemoglobina reducida son bastante diferentes en las dos longitudes de onda (Figura 3).” [5]

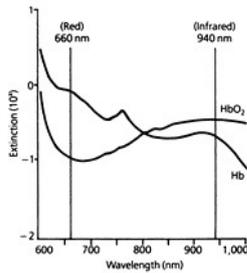


Figura 3: Espectro de absorción de la hemoglobina reducida (Hb) y oxihemoglobina (HbO<sub>2</sub>)

Luego comparando la relación de absorción de la sangre en estas dos longitudes de onda, se llega al valor deseado del porcentaje de saturación de oxígeno en sangre.

#### Modulo rastreador de movimiento y temperatura:

Se eligió el módulo MPU-6050 que es un dispositivo rastreador de movimiento (MotionTracking) de 6 ejes que combina un giroscopio y un acelerómetro de 3 ejes, sumado a un procesador digital de movimiento (DPM). Incorpora además un sensor de temperatura.[6]

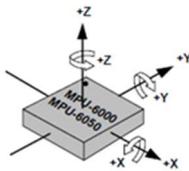


Figura 4: Orientación de Ejes de Sensibilidad y Polaridad de rotación

#### Modulo GPS:

Se utilizó un modulo GPS, basado en el chip NEO-6, con un motor de adquisición dedicado, con 2 millones de correlacionadores, capaz de realizar búsquedas masivas paralelas de tiempo/frecuencia en el espacio permitiéndole encontrar satélites al instante. El modulo además suprime las fuentes de interferencias y mitiga los efectos de trayectorias múltiples. El modulo, posee un LED testigo que enciende y parpadea cada vez que encuentra al menos 4 satélites, los datos de geo-localización son transmitidos con el protocolo NMEA. [7]

#### **b) Armado del prototipo y conexión inalámbrica con la base de datos:**

Una vez diseñado el prototipo, se procedió a programar el código en lenguaje C/C++ utilizando el IDE Arduino para ESP32. Básicamente el código programado, se encarga de relevar los datos

mencionados y acondicionarlos para ser enviados en forma inalámbrica a la web, donde pueden ser visualizados en tiempo real mediante gráficos y widgets configurables. Luego pueden ser guardados en una planilla del tipo Excel.

Esta planilla luego puede ser procesada con inteligencia artificial, y generar eventos o alarmas que avisen al operador. Para este proyecto se eligió el sitio web ThingSpeak para proyectos IOT.

Esta página es una de plataforma de análisis de datos IoT de MathWorks®.

El sitio funciona con un sistema gratuito que es suficiente para el proyecto, que incluye mensajes, y frecuencia de actualización limitados. En caso de ser necesario también hay un sistema arancelado para acceder a más beneficios.

Según se puede leer en su página de inicio:

1. ThingSpeak permite agregar, visualizar y analizar flujos de datos en vivo en la nube.
2. Algunas de las capacidades clave de ThingSpeak incluyen la capacidad de:
3. Configurar fácilmente dispositivos para enviar datos a ThingSpeak utilizando protocolos populares de IoT.
4. Visualizar los datos de los sensores en tiempo real.
5. Agregar datos bajo demanda de fuentes de terceros.
6. Utilizar el poder de MATLAB para darle sentido a los datos de IoT.
7. Ejecutar análisis de los datos IoT automáticamente según horarios o eventos.
8. Crear prototipos y construir sistemas de IoT sin configurar servidores ni desarrollar software web.
9. Actuar automáticamente sobre los datos y comunicarse utilizando servicios de terceros como Twilio® o Twitter®.”[8]

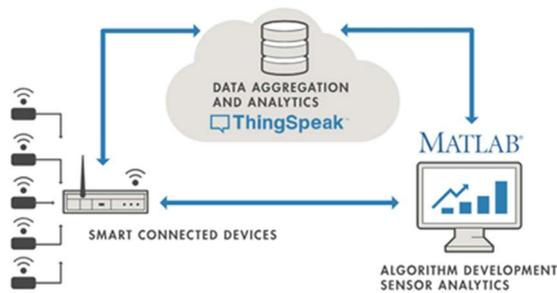


Figura 5: Sistema IoT conectado a ThingSpeak

### c) Puesta en marcha y pruebas

El sistema se ensayó sobre un Protoboard, con la finalidad de poder ajustar el prototipo. El dispositivo inicia con la medición del pulso cardiaco y oxígeno en sangre.

Para el prototipo la medición se realiza acercando el dedo índice al sensor correspondiente.

A continuación, el sistema releva los datos de temperatura (ambiente o corporal dependiendo de cuan cerca este el sujeto del sensor), datos de posición postural (ejes x, y, z ) y datos de geo-localización.

El ciclo se repite cada 15 segundos (este tiempo se puede modificar en la programación).

Siendo 15 segundos el tiempo mínimo entre envíos de mediciones al sitio web elegido en su versión gratuita.

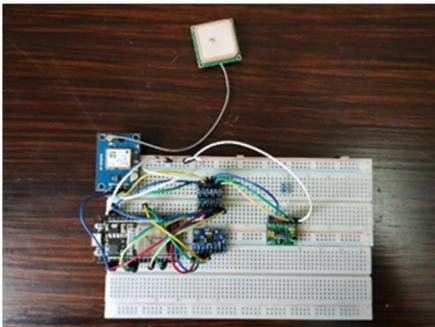


Figura 6: Prototipo dispositivo SIMAB

El prototipo fue chequeado y probado conectándolo a una computadora para visualizar los ciclos de medición.

A continuación se muestra una corrida del sistema para una medición real:

17:40:25.271 -> Inicializando Pulsoxímetro...Ok!  
17:40:25.322 -> Acerque el dedo al sensor para realizar la medición...  
17:42:38.040 -> Buscando pulso!  
17:42:38.994 -> Buscando pulso!  
17:42:39.328 -> Medición exitosa!  
17:42:39.328 -> Inicializando Acelerómetro..Ok!  
17:42:39.628 -> Aceleración X: -1.11, Y: 0.72, Z: 9.70 m/s<sup>2</sup>  
17:42:39.661 -> Rotación X: 0.05, Y: 0.02, Z: 0.00 rad/s  
17:42:39.724 -> Temperatura: 25.66 °C  
17:42:39.724 ->  
17:42:39.724 -> Inicializando GPS..OK!  
17:42:39.757 -> Buscando GPS...  
17:42:40.376 -> CHARS=242 SENTENCES=1 CSUM ERR=1  
17:42:40.408 -> Latitud/Longitud: -34.65120, -58.77571  
17:42:40.440 -> Fecha: 26/10/2023 Hora: 20:42:40.0  
17:42:40.472 -> Altitud (metros): 49.00  
17:42:40.504 -> Rumbo (grados): 0.00  
17:42:40.536 -> Velocidad (Km/h): 1.04  
17:42:40.568 -> Satélites: 4  
17:42:40.568 ->  
17:42:40.599 -> Datos que se enviaran a Thingspeak:  
17:42:40.632 -> Pulso Cardíaco: 66.56bpm / Oxígeno en sangre: 95%  
17:42:40.664 -> Aceleración Eje X: -1.11, Eje Y: 0.72, Eje Z: 9.70 m/s<sup>2</sup>  
17:42:40.727 -> Temperatura: 25.66 °C  
17:42:40.760 -> Posición según GPS: Latitud/Longitud: -34.65120, -58.77571  
17:42:40.824 ->  
17:42:40.832 -> Verificando conexión WiFi  
17:42:40.933 -> Conectando a WiFi...REDSIMAB  
17:42:40.971 -> .....  
17:42:42.967 -> Conectado a: REDSIMAB  
17:42:42.967 -> IP address: 192.168.137.64  
17:42:44.480 -> Datos enviados a ThingSpeak!  
17:42:44.512 -> Aguarde 15 segundos

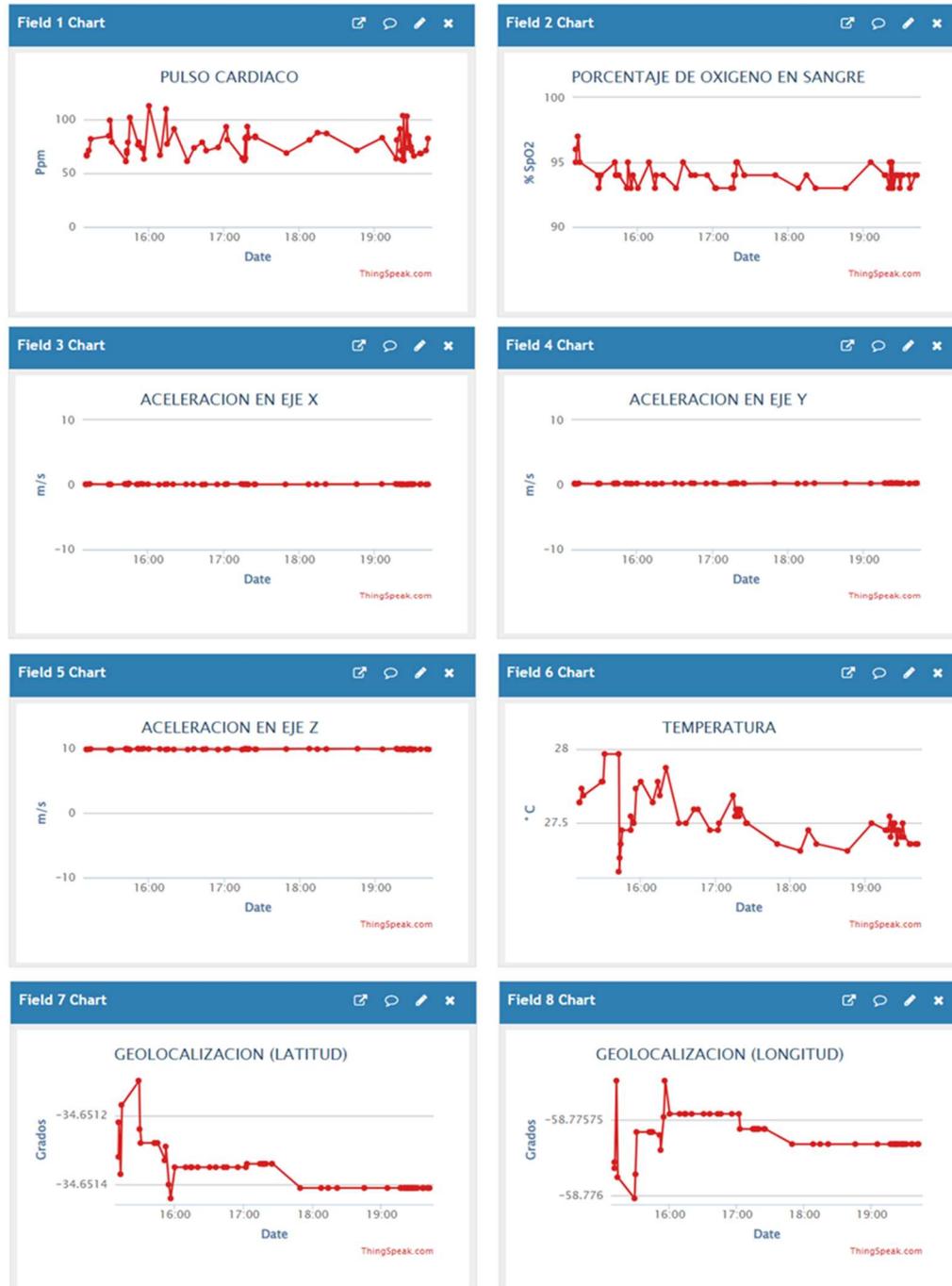
---

Como se mencionó los datos relevados son enviados al sitio web, donde son almacenados.

A continuación una captura de pantalla del sitio web mostrado sucesivas mediciones tomadas en un sujeto durante 5 hs

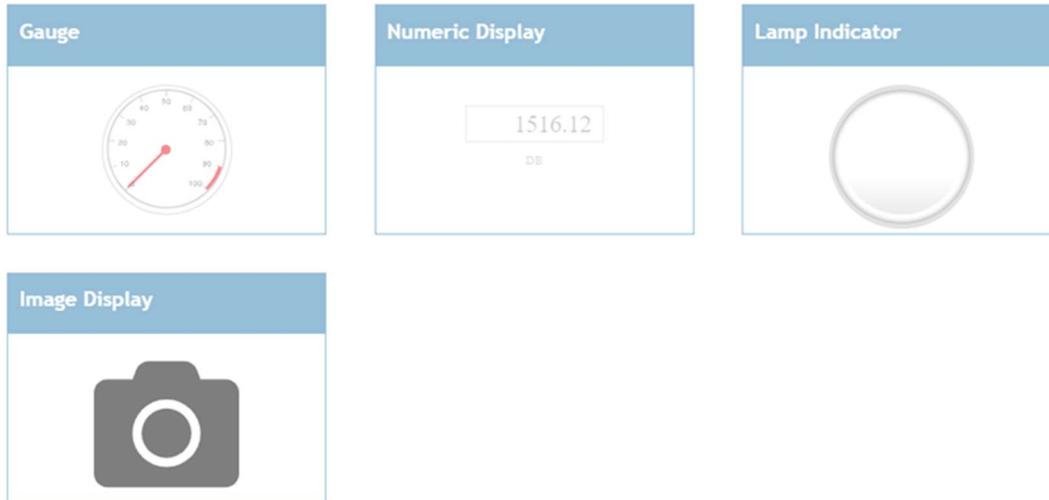
### Channel Stats

Created: [about a month ago](#)  
Last entry: [7 minutes ago](#)  
Entries: 75



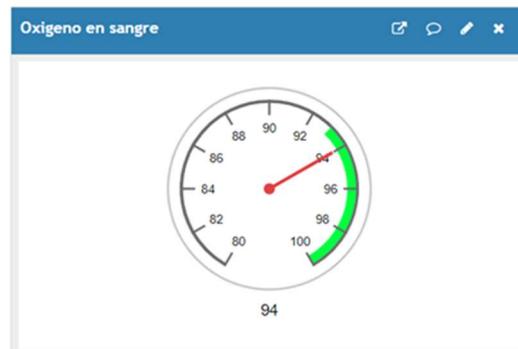
Además de los gráficos se configuraron algunos widgets:

Click on a widget to add it to the Channel ×



Next

Cancel



d) Código de programación del dispositivo:

```
//LIBRERIAS INCLUIDAS
#include <Wire.h>
#include <MAX30100_PulseOximeter.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_MPU6050.h>
#include <TinyGPS.h>
#include <HardwareSerial.h>
#include <WiFi.h>
#include <ThingSpeak.h>
#include <Arduino.h>

//VARIABLES DECLARADAS
//Pulsoximetro por I2C default
bool pulso_encontrado = false;
bool sin_dato;
float dato_pulso;
uint8_t dato_oxigeno;
//Definir valores minimos y maximos permitidos para evitar
//lecturas erroneas
#define cota_pulso_min 60
#define cota_pulso_max 120
#define cota_oxigeno_min 90
#define cota_oxigeno_max 100

//Acelerometro en otro puerto I2C
#define I2C_SCL 32
#define I2C_SDA 33
bool status_MPU;
float dato_aceleracion_x;
float dato_aceleracion_y;
float dato_aceleracion_z;
float dato_temperatura;

//GPS
const byte RXD2 = 16;
const byte TXD2 = 17;
//Variables para la obtención de datos
int year;
byte month, day, hour, minute, second, hundredths;
unsigned long chars;
unsigned short sentences, failed_checksum;
static const uint32_t GPSBaud = 4800;
float dato_latitud;
float dato_longitud;
```

```
//WiFi
const char *ssid = "REDSIMAB";
const char *password = "PaswSIMAB";

//thingspeak.com
unsigned long channelID = 2273668;
const char* WriteAPIKey ="HCTU2Q63VCEF4UQ3";

//CODIGO
//Pulsoximetro
PulseOximeter pox;

// Llamada a funcion deteccion de pulso cardiaco
void onBeatDetected() {
  Serial.println("Buscando pulso!");
  pulso_encontrado = true;
}

void configureMax30100() {
  pox.setIRLedCurrent(MAX30100_LED_CURR_50MA);
  pox.setOnBeatDetectedCallback(onBeatDetected);
}

void JM_inicio_pulsoximetro() {
  Serial.println("");
  Serial.print("Iniciando Pulsoximetro...");
  if (!pox.begin()) {
    Serial.println("Falla..Reinicie el equipo..");
    for(;;);
  } else {
    Serial.println("Ok!");
  }
}

void JM_medicion_pulsoximetro() {
  Serial.println("Acerque el dedo al sensor para realizar la medicion...");
  sin_datos = true;
  while (sin_datos) {
    pox.update();
    dato_pulso = pox.getHeartRate();
    dato_oxigeno = pox.getSpO2();
    if (pulso_encontrado && dato_pulso >= cota_pulso_min && dato_pulso <= cota_pulso_max &&
    dato_oxigeno >= cota_oxigeno_min && dato_oxigeno <= cota_oxigeno_max) {
      Serial.println("Medicion exitosa!");
      pulso_encontrado = false;
      sin_datos = false;
    }
  }
}
```

---

```
}  
}  
  
//Acelerometro  
Adafruit_MPU6050 mpu;  
TwoWire I2C_MPU = TwoWire(0);  
  
void JM_inicio_mpu6050() {  
  Serial.print("Inicializando Acelerometro..");  
  I2C_MPU.begin(I2C_SDA, I2C_SCL, 100000);  
  status_MPU = mpu.begin(0x68, &I2C_MPU);  
  //Prueba de inicializacion  
  if (!status_MPU) {  
    Serial.println("Falla..Reinicie el equipo..");  
    while (1) {  
      delay(10);  
    }  
  }  
  //configuracion del acelerometro  
  mpu.setAccelerometerRange(MPU6050_RANGE_8_G);  
  mpu.setGyroRange(MPU6050_RANGE_500_DEG);  
  mpu.setFilterBandwidth(MPU6050_BAND_5_HZ);  
  Serial.println("Ok!");  
}  
  
void JM_medicion_mpu6050() {  
  sensors_event_t a, g, temp;  
  mpu.getEvent(&a, &g, &temp);  
  
  Serial.print("Aceleracion X: ");  
  Serial.print(a.acceleration.x);  
  Serial.print(", Y: ");  
  Serial.print(a.acceleration.y);  
  Serial.print(", Z: ");  
  Serial.print(a.acceleration.z);  
  Serial.println(" m/s^2");  
  Serial.print("Rotacion X: ");  
  Serial.print(g.gyro.x);  
  Serial.print(", Y: ");  
  Serial.print(g.gyro.y);  
  Serial.print(", Z: ");  
  Serial.print(g.gyro.z);  
  Serial.println(" rad/s");  
  Serial.print("Temperatura: ");  
  Serial.print(temp.temperature);  
  Serial.println(" °C");  
  Serial.println("");  
}
```

```
dato_aceleracion_x = a.acceleration.x;
dato_aceleracion_y = a.acceleration.y;
dato_aceleracion_z = a.acceleration.z;
dato_temperatura = temp.temperature;
}

//GPS
HardwareSerial neogps(2);
TinyGPS gps;

void JM_inicio_gps() {
  neogps.begin(9600, SERIAL_8N1, TXD2, RXD2);
  Serial.println("Inicializando GPS..OK!");
}

void JM_medicion_gps() {
  Serial.println("Buscando GPS...");
  bool newData = false;
  unsigned long chars;
  unsigned short sentences, failed;

  while (!newData) {
    for (int espera=0; espera<=1500; espera++) {
      if (neogps.available()) {
        char c = neogps.read();
        //Serial.write(c); // Descomente esta línea si desea ver los datos del GPS fluyendo.
        if (gps.encode(c)) { // Llegó una nueva sentencia válida?
          newData = true;
          espera=1500;
        }
      }
    }
    newData = true;
  }
}

if (newData) {
  newData = false;
  float flat, flon;
  unsigned long age;
  gps.f_get_position(&flat, &flon, &age);
  gps.stats(&chars, &sentences, &failed);
  Serial.print(" CHARS=");
  Serial.print(chars);
  Serial.print(" SENTENCES=");
  Serial.print(sentences);
  Serial.print(" CSUM ERR=");
  Serial.println(failed);
```

---

```
float latitude, longitude;
gps.f_get_position(&latitude, &longitude);
Serial.print("Latitud/Longitud: ");
Serial.print(latitude,5);
Serial.print(", ");
Serial.println(longitude,5);
gps.crack_datetime(&year, &month, &day, &hour, &minute, &second, &hundredths);
Serial.print("Fecha: "); Serial.print(day, DEC); Serial.print("/");
Serial.print(month, DEC); Serial.print("/"); Serial.print(year);
Serial.print(" Hora: "); Serial.print(hour, DEC); Serial.print(":");
Serial.print(minute, DEC); Serial.print(":"); Serial.print(second, DEC);
Serial.print("."); Serial.println(hundredths, DEC);
Serial.print("Altitud (metros): ");
Serial.println(gps.f_altitude());
Serial.print("Rumbo (grados): "); Serial.println(gps.f_course());
Serial.print("Velocidad (km/h): ");
Serial.println(gps.f_speed_kmph());
Serial.print("Satélites: "); Serial.println(gps.satellites());
Serial.println();
gps.stats(&chars, &sentences, &failed_checksum);
dato_latitud = latitude;
dato_longitud = longitude;
}
if (chars == 0)
  Serial.println("*** No se reciben caracteres del GPS: verifique el sistema ***");
}

//Relevo de datos a enviar
void JM_relevo_datos() {
  Serial.println();
  Serial.println("Datos que se enviaran a Thingspeak:");
  Serial.print("Pulso Cardiaco:");
  Serial.print(dato_pulso);
  Serial.print("bpm / Oxigeno en sangre:");
  Serial.print(dato_oxigeno);
  Serial.println("%");
  Serial.print("Aceleracion Eje X: ");
  Serial.print(dato_aceleracion_x);
  Serial.print(", Eje Y: ");
  Serial.print(dato_aceleracion_y);
  Serial.print(", Eje Z: ");
  Serial.print(dato_aceleracion_z);
  Serial.println(" m/s^2");
  Serial.print("Temperatura: ");
  Serial.print(dato_temperatura);
  Serial.println(" °C");
  Serial.print("Posicion segun GPS: Latitud/Longitud: ");
  Serial.print(dato_latitud,5);
```

---

```
Serial.print(", ");
Serial.println(dato_longitud,5);
Serial.println();
}

//Conexion a WiFi
void JM_conectar_wifi() {
  Serial.print("Conectando a WiFi...");
  Serial.println(ssid);
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(100);
    Serial.print('.');
  }

  Serial.println("");
  Serial.print("Conectado a:\t");
  Serial.println(ssid);
  Serial.print("IP address:\t");
  Serial.println(WiFi.localIP());
}

//Envio de datos a thingspeak.com
WiFiClient cliente;

void JM_envio_datos_thingspeak() {
  ThingSpeak.begin(cliente);
  ThingSpeak.setField(1,dato_pulso);
  ThingSpeak.setField(2,dato_oxigeno);
  ThingSpeak.setField(3,dato_aceleracion_x);
  ThingSpeak.setField(4,dato_aceleracion_y);
  ThingSpeak.setField(5,dato_aceleracion_z);
  ThingSpeak.setField(6,dato_temperatura);
  ThingSpeak.setField(7,dato_latitud);
  ThingSpeak.setField(8,dato_longitud);
  ThingSpeak.writeFields(channelID,WriteAPIKey);
  Serial.println("Datos enviados a ThingSpeak!");
  Serial.println("Aguarde 15 segundos");
  delay(15000);
}

//Codigo principal
void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
  digitalWrite(LED_BUILTIN, HIGH);
  Serial.begin(9600);
```

---

```
JM_inicio_pulsoximetro();  
_____  
configureMax30100();  
JM_medicion_pulsoximetro();  
digitalWrite(LED_BUILTIN, LOW);  
JM_inicio_mpu6050();  
JM_medicion_mpu6050();  
JM_inicio_gps();  
JM_medicion_gps();  
JM_relevo_datos();  
}  
  
void loop() {  
  Serial.println("Verificando conexion Wifi");  
  if (WiFi.status() != WL_CONNECTED) {  
    JM_conectar_wifi();  
  }  
  JM_envio_datos_thingspeak();  
  setup ();  
}
```

#### ***4. Resultados y productos***

---

De la investigación se desprende la necesidad de seguir profundizando lo aprendido y aparecen las siguientes acciones a desarrollar en un ulterior proyecto:

- Entrenar a la red neuronal con información real de usuarios “sanos” desde los sensores
- Cargar y entrenar a la red neuronal con diferentes historias clínicas
- Realizar una integración del hardware en un dispositivo “minimizado”. Aparece la oportunidad de integrar al futuro proyecto a la carrera de Diseño Industrial.

#### ***5. Impacto***

*- Describir el impacto del proyecto en relación a la problemática planteada*

---



Prof. Ing. Pedro M. Giuffrida

**Fecha:** 28 / 02 / 2024

**Firma Responsable del Área de la UNM donde radica el Proyecto:**

---