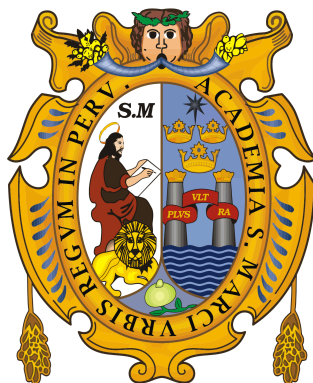


**UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS**

**FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**

**ESCUELA PROFESIONAL ACADÉMICA DE INGENIERÍA DE SOFTWARE**



**Proyecto de Monitoreo de Pulso y Saturación de Oxígeno**

**Docentes:** Herrera Quispe, José Alfredo

Rosas Cuevas, Yessica

**Curso:** Internet de las cosas

**Sección:** 3

**Integrantes:**

Hurtado Santos, Estiven Salvador 20200135

López Terrones, Xiomy Ximena 20200020

Mondragón Zúñiga, Rubén Alberto 20200082

Morales Robladillo, Nicole Maria 20200136

**Semestre 2024 - 1**

# **Proyecto de Monitoreo de Pulso y Saturación de Oxígeno**

## **1. Introducción**

La salud cardiovascular es un tema de vital importancia en el ámbito médico y tecnológico. En la actualidad, millones de personas padecen enfermedades relacionadas con la presión arterial y los niveles de oxígeno en sangre. Según la Organización Mundial de la Salud (OMS), las enfermedades cardiovasculares son la principal causa de muerte en todo el mundo, cobrando la vida de aproximadamente 17.9 millones de personas al año .

El monitoreo constante de estos parámetros vitales es esencial para la prevención y manejo de estas enfermedades. En este contexto, hemos desarrollado un sensor innovador que mide de manera precisa y en tiempo real los niveles de oxígeno y presión arterial. Este dispositivo no solo es crucial para pacientes con condiciones crónicas, sino también para cualquier persona que desee mantener un control riguroso de su salud.

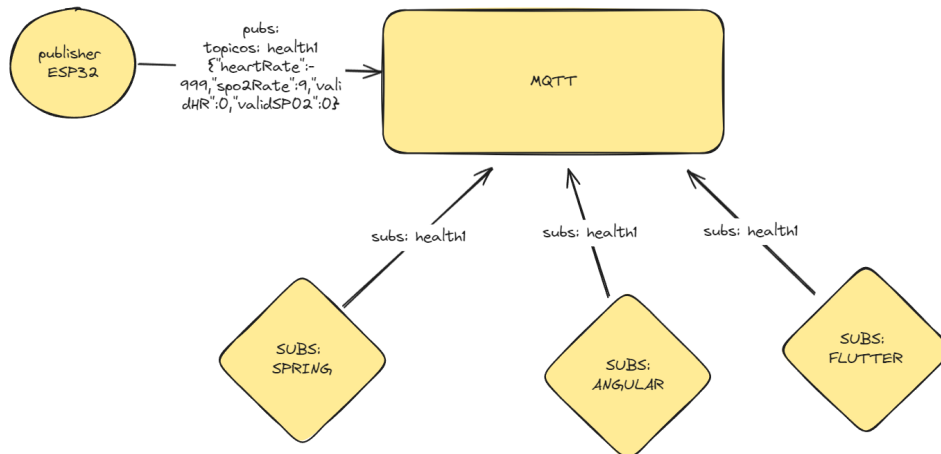
Con este sensor, esperamos contribuir a la reducción de las estadísticas alarmantes relacionadas con las enfermedades cardiovasculares, proporcionando una herramienta que permita la detección temprana y la intervención oportuna.

## **2. Descripción**

Este sistema monitorea el nivel de oxígeno en la sangre del paciente en cuestión. La lectura que recibirá el sensor MAX 30100 que estará conectado a una placa ESP32, conectada a una red WIFI, para enviar datos a un servidor MQTT. Múltiples aplicaciones pueden suscribirse a MQTT, en este caso, una aplicación Spring que guardará las lecturas en una base de datos MongoDB, una aplicación Angular que mostrará el valor de la lectura de Saturación y pulso y un aplicativo Flutter. Una vez que se tenga esta información del cliente por medio del oxímetro, este estará asociado a un médico el cual podrá monitorizar el estado del paciente en cuanto a los niveles de oxígeno y si este está con estado crítico, el médico recibirá una alerta con la información del cliente que se encuentra con malos niveles de oxígeno para que agende una cita.

Este proyecto servirá para que los médicos tengan una facilidad de control de sus pacientes en particular de los que tienen problemas respiratorios o de presión tanto alta como baja para darles un seguimiento más exhaustivo y que los pacientes se sientan más seguros. A su vez, con la data histórica, en un futuro crear un modelo predictivo que permita prevenir un posible desastre con el cliente en caso los patrones de salud lo reflejan.

A continuación un modelo en MockUp de cómo es que sería a nivel borrador las conexiones:



### 3. PROBLEMÁTICA

A pesar de los avances en la tecnología médica, el control adecuado de la presión arterial y los niveles de oxígeno en sangre sigue siendo un desafío para muchas personas. Las estadísticas de la OMS revelan que las enfermedades cardiovasculares siguen siendo la principal causa de muerte a nivel mundial.

Los dispositivos de monitoreo tradicionales, a menudo caros y complicados de usar, no están accesibles para todos. Esta falta de accesibilidad y facilidad de uso impide que muchas personas mantengan un control constante de su salud, lo que puede llevar a diagnósticos tardíos y complicaciones graves.

En resumen, la problemática que abordamos es:

- **Alta prevalencia de enfermedades cardiovasculares:** Con 17.9 millones de muertes anuales.
- **Accesibilidad limitada a dispositivos de monitoreo:** Los dispositivos actuales son costosos y complicados.
- **Necesidad de detección temprana:** La falta de monitoreo constante impide intervenciones oportunas.

- **Disponibilidad de la información en tiempo real:** Necesidad de tener un seguimiento por parte del médico a distancia de su paciente.

#### **4. MOTIVACIÓN**

Lo que nos motiva a desarrollar este sensor de oxígeno y presión arterial es la convicción de que podemos hacer una diferencia real en la vida de las personas. Todos conocemos a alguien que ha sido afectado por problemas de salud, ya sea en nuestra familia o amigos. Queremos que esta tecnología sea accesible para todos, no solo para aquellos que pueden pagar dispositivos caros o complicados. Imaginemos un mundo donde cualquiera pueda monitorear su salud de manera sencilla y en tiempo real, previniendo problemas graves antes de que ocurran, también poder notificar al médico encargado en tiempo real. Nos emociona pensar en el impacto positivo que nuestro proyecto puede tener, ayudando a muchas personas a llevar una vida más saludable y tranquila. Este sensor no es solo un dispositivo, es nuestra manera de contribuir a un mundo mejor.

#### **5. OBJETIVOS**

- a) Implementar el sensor de oxígeno y presión:** Desarrollar un dispositivo preciso y confiable que mida en tiempo real los niveles de oxígeno en sangre y la presión arterial.
- b) Disponer de los datos capturados en la nube:** Asegurar que los datos recogidos por el sensor se almacenen y gestionan en la nube, facilitando su acceso y manipulación a través de dispositivos móviles.

**c) Contribuir al control de la salud:** Proveer una herramienta accesible que permita a los usuarios monitorear y gestionar su salud de manera proactiva, contribuyendo a la detección temprana y prevención de enfermedades cardiovasculares.

## **6. PROPUESTA**

Nuestra propuesta consiste en desarrollar un innovador sensor de oxígeno y presión arterial que sea accesible y fácil de usar. Este dispositivo no sólo medirá estos parámetros vitales en tiempo real, sino que también enviará los datos capturados a la nube. De esta manera, los usuarios podrán acceder y gestionar su información de salud desde cualquier dispositivo móvil. Con esta herramienta, buscamos empoderar a las personas para que puedan monitorear su salud de manera continua, detectar posibles problemas de forma temprana y tomar decisiones informadas sobre su bienestar. Creemos que, al facilitar el control de la salud a través de la tecnología, podemos ayudar a reducir las tasas de enfermedades cardiovasculares y mejorar la calidad de vida de muchas personas.

## **7. REQUERIMIENTOS DEL SISTEMA**

### **ESP32:**

El ESP32 es un microcontrolador potente y versátil. Es capaz de conectarse a Wifi y Bluetooth.

### **Uso en el proyecto:**

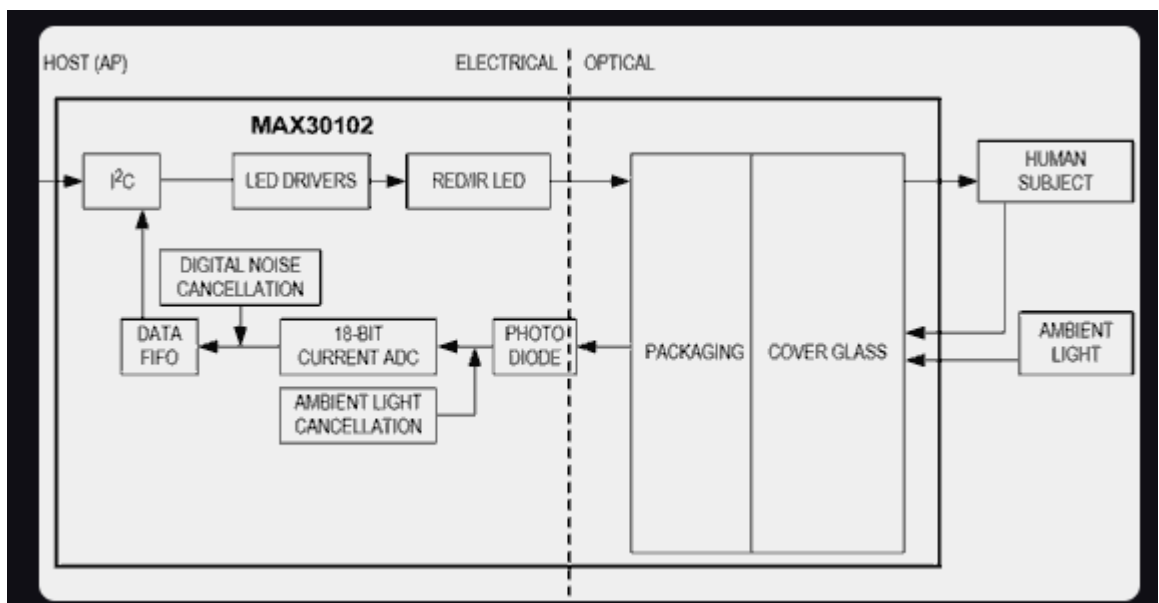
- **Conexión con el Sensor:** Se conecta con el sensor MAX30100 mediante protocolos como I2C para recibir datos de oxígeno en sangre y ritmo cardíaco.

- **Procesamiento de Datos:** Procesa los datos obtenidos para determinar los niveles de oxígeno y ritmo cardíaco.

## MAX 30102

El MAX 30100 es un sensor de oxímetro y pulso integrado desarrollado por Maxim Integrated. Es ideal para aplicaciones de monitoreo de salud debido a su precisión y bajo consumo de energía.

El MAX30102 incorpora dos LED, uno de espectro rojo (660 nm) y otro de infrarrojo (880 nm), así como fotodiodos para medir la luz reflejada y un ADC de 18 bits y frecuencia de muestreo de 50 sps (samples per second) a 3200sps.

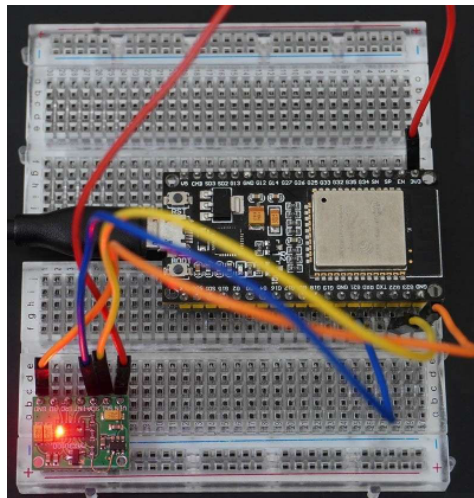


## Uso en el proyecto:

- **Medición:** El sensor MAX30102 mide continuamente los niveles de oxígeno en la sangre y el ritmo cardíaco del paciente.
- **Transmisión de Datos:** Envía los datos al ESP32 a través de la interfaz I2C.

- **Almacenamiento y Análisis:** Los datos pueden ser almacenados y analizados para detectar patrones o eventos críticos en la base de datos Mongo.
- **Alertas:** Si los datos indican una condición crítica, el Servidor activa un sistema de alertas para notificar al paciente y al médico.

#### **Conexión del circuito:**



## **8. ARQUITECTURA**

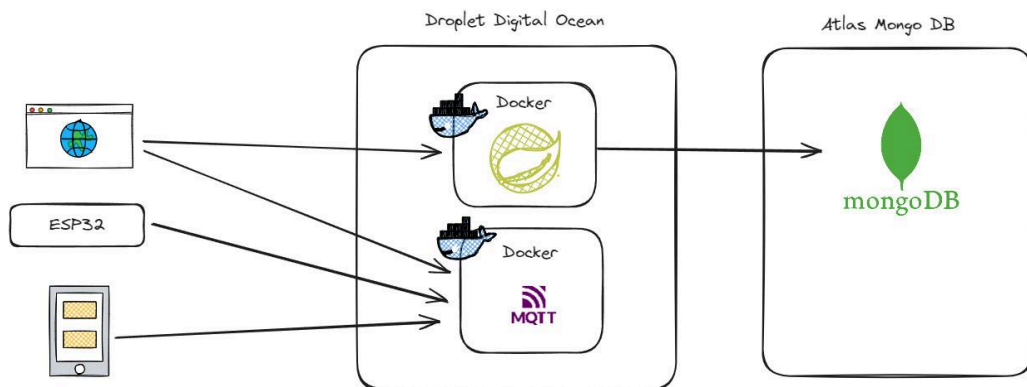
Para la arquitectura del sistema, se implementó diversas tecnologías que permiten la funcionalidad del sistema. Se usó MQTT como mecanismo de mensaje para enviar las mediciones de las lecturas a un servidor Mosquitto. Este servidor permitirá a otros aplicativos como los desarrollados para poder realizar los tratamientos de los datos.

#### **La arquitectura del sistema se compone de lo siguiente:**

- App web Angular
- Aplicativo móvil Flutter
- MQTT Server



- Spring Boot App
- MongoDB.



## 9. RESOLUCIÓN DE PROBLEMA

**9.1. Solución:** El sistema propuesto resuelve el problema del monitoreo continuo y en tiempo real de los niveles de oxígeno en sangre y el ritmo cardíaco mediante el uso de sensores, microcontroladores, comunicación eficiente y almacenamiento y procesamiento robustos de datos.

### 9.2. Detalles de la Solución

#### 9.2.1. Desarrollo del Dispositivo.

Se utilizó el microcontrolador ESP32, el sensor MAX30100 y componentes como resistencias y cables de conexión.

##### 9.2.1.1. Conexión I2C entre ESP32 y Sensor MAX30100.

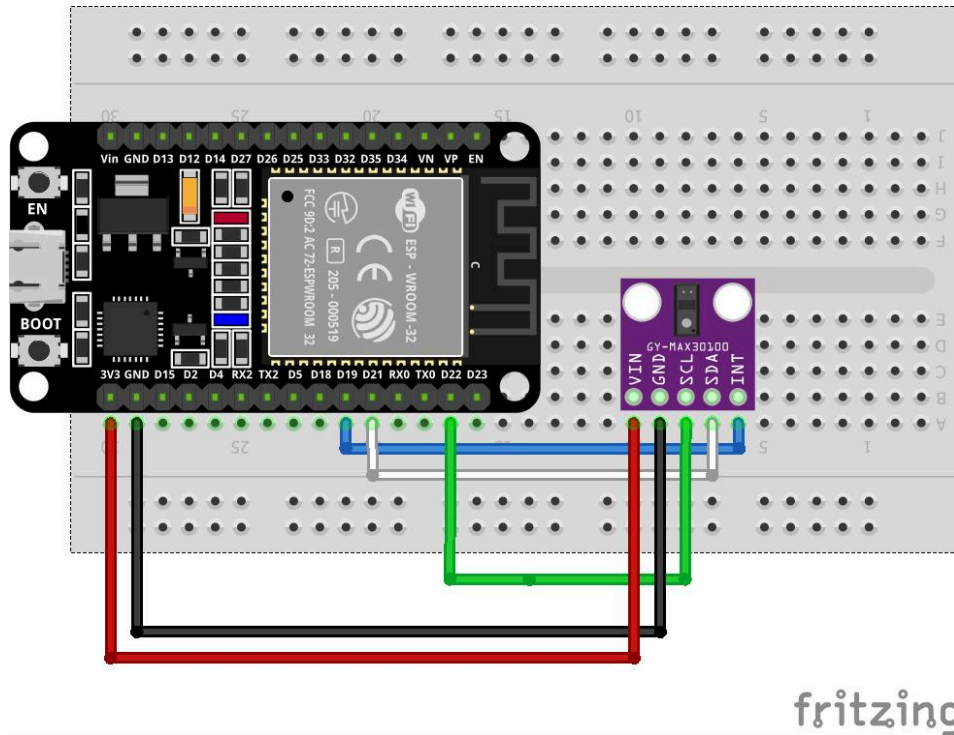
El I2C (Inter-Integrated Circuit) es el protocolo de comunicación serial que utiliza dos líneas para la transmisión de datos: SDA (Serial Data) y SCL (Serial Clock)

#### Pines de Conexión:

- **SDA (Data):** Conectar el pin SDA del sensor MAX30100 al pin GPIO21 (SDA) del ESP32.

- **SCL (Clock):** Conectar el pin SCL del sensor MAX30100 al pin GPIO22 (SCL) del ESP32.
- **VCC:** Conectar el pin VCC del sensor MAX30100 al pin 3.3V del ESP32.
- **GND:** Conectar el pin GND del sensor MAX30100 al pin GND del ESP32.

**Diagrama de conexión:**



## 9.2.2. Desarrollo del Programa.

### 9.2.2.1. Establecer Docker con la Imagen de Eclipse Mosquito

Para configurar el servidor MQTT utilizando Docker con la imagen de Eclipse Mosquitto, sigue estos pasos:

**Instalación de Docker:** Se ejecutaron los comandos necesarios para instalar Docker en el sistema operativo.

```
sudo apt update
sudo apt install docker.io
```

**Descarga de la Imagen de Eclipse Mosquitto:** Utilizamos el siguiente comando para descargar la imagen oficial de Eclipse Mosquitto desde Docker Hub.

```
sudo docker pull eclipse-mosquitto
```

**Ejecución del Contenedor:** Iniciar un contenedor Docker basado en la imagen de Eclipse Mosquitto. Mapeamos los puertos necesarios para la comunicación MQTT.

```
sudo docker run -d --name mqtt-broker -p 1883:1883 -p 9001:9001 eclipse-mosquitto
```

-d: Ejecuta el contenedor en segundo plano (modo daemon).

--name mqtt-broker: Asigna un nombre al contenedor para facilitar su gestión.

-p 1883:1883 -p 9001:9001: Mapea el puerto 1883 para conexiones TCP MQTT y el puerto 9001 para conexiones WebSocket.

**Ejecutar el Programa y configurar el SP32 como Publicador:** Una vez que el servidor MQTT está configurado y los puertos están abiertos, se procede a ejecutar el programa en el ESP32 para que actúe como un publicador de datos.

**Conectar a Wi-Fi y Enviar Datos:** Después de cargar el programa en el ESP32 y asegurar que se conecte correctamente a la red Wi-Fi, implementa la lógica para leer los datos del sensor (MAX30100) y enviarlos al servidor MQTT.

```
etch_jul10a.ino
1  #include <Wire.h>
2  #include <WiFi.h>
3  #include "MAX30105.h"
4  #include "spo2_algorithm.h"
5  #include "heartRate.h"
6  #include <WiFi.h>
7  #include <PubSubClient.h>
8  #include <ArduinoJson.h>
9  #include <ArduinoJson.hpp>
10 // WiFi
11 const char *ssid = "Moto g50"; // Enter your Wi-
12 const char *password = "soypobre"; // Enter Wi-Fi pa
13
14 // MQTT Broker
15 const char *mqtt_broker = "64.23.217.53";
16 const char *topic = "health1";
17 const int mqtt_port = 1883;
18
19 MAX30105 particleSensor;
```

## 10. RESULTADOS

Como resultados tenemos que nuestro sistema "Sensor de presión y oxígeno" es una gran opción debido a su practicidad de tener seguimiento con respecto a su salud cardiovascular. Además que

brinda una posibilidad para tener un seguimiento histórico por medio de un dashboard. También, comentar su alta disponibilidad tanto móvil como web para el seguimiento por parte del médico a su paciente.

Cabe mencionar que se tiene planeado implementar un apartado de Inteligencia Artificial que permita dar un diagnóstico previo de acuerdo a los parámetros obtenidos de la presión y el oxígeno del paciente para evaluar si es “Positivo” o “Negativo” en un diagnóstico para COVID.

### **Desarrollo de Modelo ML:**

Para el desarrollo de nuestro modelo de Machine Learning usamos de base el dataset de kaggle “Covid-19 Temperature, Oxygen & Pulse Rate readings” que nos brinda una densidad de data de 10 mil registros con columnas “ID”, “Oxygen”, “PulseRate”, “Temperature” y “Result”.

```
qt_dataset.csv
```

```
dataset > qt_dataset.csv > data
```

|       | ID   | Oxygen | PulseRate | Temperature | Result   |
|-------|------|--------|-----------|-------------|----------|
| 1     | 9974 | 90     | 100       | 102         | Positive |
| 9977  | 9975 | 93     | 48        | 95          | Negative |
| 9978  | 9976 | 85     | 87        | 100         | Positive |
| 9979  | 9977 | 94     | 118       | 95          | Negative |
| 9980  | 9978 | 88     | 52        | 101         | Positive |
| 9981  | 9979 | 94     | 108       | 101         | Positive |
| 9982  | 9980 | 88     | 93        | 100         | Positive |
| 9983  | 9981 | 90     | 91        | 97          | Negative |
| 9984  | 9982 | 95     | 43        | 95          | Negative |
| 9985  | 9983 | 85     | 56        | 100         | Positive |
| 9986  | 9984 | 87     | 71        | 102         | Positive |
| 9987  | 9985 | 87     | 125       | 101         | Positive |
| 9988  | 9986 | 98     | 110       | 95          | Negative |
| 9989  | 9987 | 92     | 68        | 96          | Negative |
| 9990  | 9988 | 93     | 42        | 97          | Negative |
| 9991  | 9989 | 90     | 116       | 103         | Positive |
| 9992  | 9990 | 93     | 62        | 96          | Negative |
| 9993  | 9991 | 95     | 116       | 95          | Negative |
| 9994  | 9992 | 95     | 41        | 96          | Negative |
| 9995  | 9993 | 98     | 125       | 100         | Negative |
| 9996  | 9994 | 100    | 42        | 96          | Negative |
| 9997  | 9995 | 95     | 124       | 97          | Negative |
| 9998  | 9996 | 88     | 70        | 100         | Positive |
| 9999  | 9997 | 99     | 56        | 105         | Negative |
| 10000 | 9998 | 92     | 49        | 98          | Positive |
| 10001 | 9999 | 85     | 52        | 99          | Positive |

Entrenamiento del modelo en base al dataset "qt\_dataset.csv". Se utiliza el Random Forest ideal para clasificar las entradas. Además que se prepara el dataset, entre las medidas se elimina la columna "Temperatura" debido a que nuestro sistema solo toma el pulso y oxígeno.

```
modelo.py X
modelo.py > ...
1 import pandas as pd
2 from sklearn.model_selection import train_test_split
3 from sklearn.preprocessing import StandardScaler
4 from sklearn.ensemble import RandomForestClassifier
5 from sklearn.svm import SVC
6 from sklearn.metrics import accuracy_score, classification_report
7
8
9 file_path = './dataset/qt_dataset.csv'
10 data = pd.read_csv(file_path)
11
12 data = data.drop(columns=['Temperature'])
13
14 data['Result'] = data['Result'].map({'Negative': 0, 'Positive': 1})
15
16 X = data[['Oxygen', 'PulseRate']]
17 y = data['Result'] #Target
18
19 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, ra
20
21 scaler = StandardScaler()
22 X_train = scaler.fit_transform(X_train)
23 X_test = scaler.transform(X_test)
24
25 rf_model = RandomForestClassifier(random_state=42)
26 rf_model.fit(X_train, y_train)
27
28 rf_y_pred = rf_model.predict(X_test)
29 rf_accuracy = accuracy_score(y_test, rf_y_pred)
30 rf_report = classification_report(y_test, rf_y_pred)
```

Implementación de una API basada en nuestro modelo. Le asignados un endpoint "POST" para que reciba 2 parámetros de "Oxígeno" y "Pulso". Para que por medio del modelo pueda predecir si estas entradas representan una salida positiva o negativa al COVID.

```

class PredictValues(BaseModel):
    oxygen: float
    pulse_rate: float

@app.post("/predict/values")
async def predict_values(values: PredictValues):
    try:
        if scaler is None:
            raise ValueError("Scaler no está cargado correctamente.")
        if rf_model is None:
            raise ValueError("Modelo de Random Forest no está cargado correctamente.")

        # Normalizar los datos de entrada
        input_data = np.array([[values.oxygen, values.pulse_rate]])
        scaled_input = scaler.transform(input_data)

        # Realizar la predicción
        prediction = rf_model.predict(scaled_input)
        result = 'Positive' if prediction[0] == 1 else 'Negative'

        return {"prediction": result}
    except Exception as e:
        raise HTTPException(status_code=500, detail=f"Error procesando los valores: {e}")

```

## Vista de la API:

**PROYECTO BASURITA** 0.1.0 OAS 3.1

/openapi.json

**Subida de Imágenes** `predict/image`

**BASURA DETECTION**

**default**

- POST** `/predict/image` Predict Image
- POST** `/predict/values` Predict Values

**POST** `/predict/values` Predict Values

Parameters

No parameters

Request body *required* `application/json`

```

{
  "oxygen": 84,
  "pulse_rate": 81
}

```

Execute Clear

Responses

Curl

```

curl -X 'POST' \
  'http://64.23.217.51/predict/values' \
  -H 'Accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "oxygen": 84,
    "pulse_rate": 81
  }'

```

Request URL

`http://64.23.217.51/predict/values`

Server response

Code Details

200

Response body

```

{
  "prediction": "Positive"
}

```

Download

## **11. CONCLUSIONES**

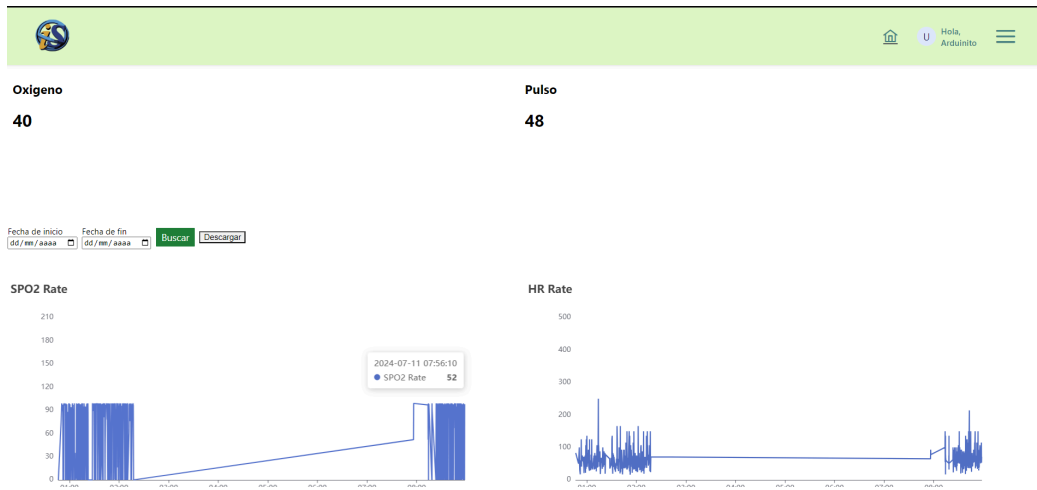
En resumen, nuestro proyecto del sensor de oxígeno y presión arterial es un gran paso adelante en el campo de la salud tecnológica. Este dispositivo no solo cubre una necesidad urgente al ofrecer monitoreo en tiempo real de estos parámetros vitales, sino que también hace que los datos sean accesibles y fáciles de manejar a través de la nube. Esto permitirá a los usuarios seguir de cerca su salud en cualquier momento y desde cualquier lugar, ayudando a detectar problemas a tiempo y tomar decisiones informadas.

Este sensor puede cambiar la forma en que la gente monitorea su salud, proporcionando una herramienta crucial para prevenir enfermedades. Queremos hacer que la tecnología avanzada de monitoreo sea accesible para todos, rompiendo las barreras económicas y técnicas.

En definitiva, nuestro proyecto no es solo una innovación tecnológica, sino también una misión para mejorar la vida de las personas y crear una sociedad más saludable y consciente. Con este sensor, esperamos hacer una diferencia real en la lucha contra las enfermedades cardiovasculares, promoviendo un futuro donde controlar la salud sea algo al alcance de todos.

## **12. ANEXOS**

Vistas del aplicativo Web

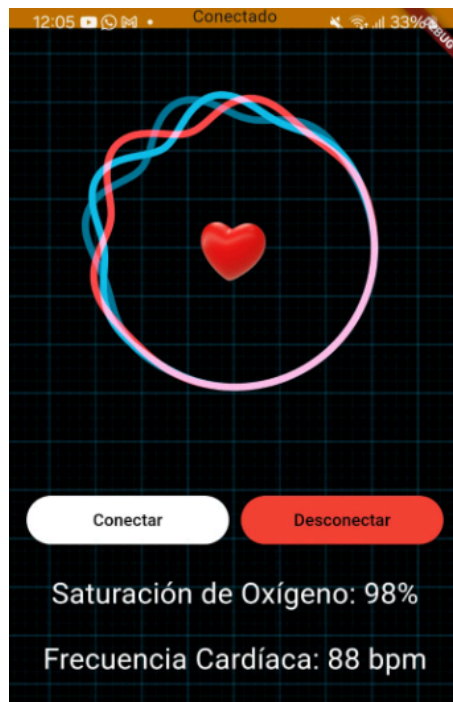


Servicios en Docker corriendo:

```
The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Thu Jul 11 13:21:25 2024 from 162.243.188.66
root@MANTENIMIENTO-G1:~# docker ps
CONTAINER ID   IMAGE               COMMAND                  CREATED        STATUS        PORTS
06b69ade5224   arduinitos-frontend "docker-entrypoint.s..." 3 hours ago    Up 2 hours    0.0.0.0:4201->4200/tcp,
:::4201->4200/tcp
6ce169c72297   arduinitos-backend  "java -jar /app/app. ...." 3 hours ago    Up 3 hours    0.0.0.0:8086->8080/tcp,
:::8086->8080/tcp
5e55d59d5891   eclipse-mosquitto   "/docker-entrypoint. ...." 2 weeks ago    Up 3 hours    0.0.0.0:1883->1883/tcp,
:::1883->1883/tcp, 0.0.0.0:9001->9001/tcp,
:::9001->9001/tcp
root@MANTENIMIENTO-G1:~#
```

Aplicativo Móvil - Vista Común



Aplicativo Móvil - Vista Alerta Oxígeno bajo





## Aplicativo Móvil – Vista Alerta Oxígeno y Frecuencia Cardíaca Baja



## Referencias Bibliográficas:

Bhojani, A. M. (2023, August 31). Implementing MQTT Protocol in Your Flutter App: A Step-by-Step Example. *Medium*.

<https://medium.com/@mohibbhojani/implementing-mqtt-protocol-in-your-flutter-app-a-step-by-step-example-d6bd57eec0ec>

Mascarenhas, R. (2021.). *COVID-19 temperature, oxygen, pulse rate* [Data set]. Kaggle.

<https://www.kaggle.com/datasets/rishanmascarenhas/covid19-temperatureoxygenpulse-rate>

World Health Organization. (2021). *Cardiovascular diseases (CVDs)*.

[https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-\(cvds\)](https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds))

Torbey, M. T. (2023). The magnitude of blood pressure reduction in intracerebral hemorrhage is associated with in-hospital outcomes [Research publication]. ResearchGate.

[https://www.researchgate.net/publication/376321892\\_The\\_Magnitude\\_of\\_Blood\\_Pressure\\_Reduction\\_in\\_Intracerebral\\_hemorrhage\\_Is\\_Associated\\_with\\_In-Hospital\\_Outcomes\\_5115](https://www.researchgate.net/publication/376321892_The_Magnitude_of_Blood_Pressure_Reduction_in_Intracerebral_hemorrhage_Is_Associated_with_In-Hospital_Outcomes_5115)