



UNIVERSIDADE FEDERAL DO AMAZONAS
FACULDADE DE TECNOLOGIA
ENGENHARIA DA COMPUTAÇÃO

VALNEY MARINHO DO NASCIMENTO JÚNIOR

SISTEMA DE MONITORAMENTO REMOTO DE BATIMENTOS CARDÍACOS E
OXIMETRIA UTILIZANDO SENSORES MAX30100 E MÓDULOS DE AQUISIÇÃO
COM ESP8266

Manaus - Amazonas

Abril de 2022

VALNEY MARINHO DO NASCIMENTO JÚNIOR

**SISTEMA DE MONITORAMENTO REMOTO DE BATIMENTOS CARDÍACOS E
OXIMETRIA UTILIZANDO SENSORES MAX30100 E MÓDULOS DE AQUISIÇÃO
COM ESP8266**

Monografia apresentada à Coordenação de Engenharia da Computação da Universidade Federal do Amazonas, como parte dos requisitos necessários à obtenção do título de Engenheiro da Computação.

Orientador: Professor. MSc. Francisco de Assis Pereira Januário.

Manaus - Amazonas

Abril de 2022

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

N244s Nascimento Júnior, Valney Marinho do
Sistema de monitoramento remoto de batimentos cardíacos e oximetria utilizando sensores MAX30100 e módulos de aquisição com ESP8266 / Valney Marinho do Nascimento Júnior . 2022
56 f.: il.; 31 cm.

Orientador: Francisco de Assis Pereira Januário
TCC de Graduação (Engenharia da Computação) - Universidade Federal do Amazonas.

1. Max30100. 2. Esp8266. 3. Raspberry. 4. Oximetria. 5. Covid-19. I. Januário, Francisco de Assis Pereira. II. Universidade Federal do Amazonas III. Título

VALNEY MARINHO DO NASCIMENTO JÚNIOR

**SISTEMA DE MONITORAMENTO REMOTO DE BATIMENTOS CARDÍACOS E
OXIMETRIA UTILIZANDO SENSORES MAX30100 E MÓDULOS DE AQUISIÇÃO
COM ESP8266**

Monografia apresentada à Coordenação do Curso de Engenharia da Computação da Universidade Federal do Amazonas, como parte dos requisitos necessários à obtenção do título de Engenheiro da Computação.

Aprovado em 26 de Abril de 2022:

Banca Examinadora

Francisco de Assis Pereira Januário

Prof. MSc. Francisco de Assis Pereira Januário - Presidente e Orientador
Universidade Federal do Amazonas - UFAM

Frederico da Silva Pinagé

Prof. Dr. Frederico da Silva Pinagé - Membro
Universidade Federal do Amazonas - UFAM

Thiago Brito

Prof. Dr. Thiago Brito Bezerra - Membro
Universidade Federal do Amazonas - UFAM

AGRADECIMENTOS

Agradeço primeiramente a Deus pelas bênçãos recebidas desde o início da minha aprovação no vestibular até o momento da conclusão do meu curso. Agradeço à Ele pela força nas inúmeras madrugadas de estudos e pelo suporte nos momentos de desânimo durante o curso. Hoje estou aqui, finalizando e podendo dizer: “Até aqui me ajudou o Senhor”.

Agradeço aos meus pais, Valney e Gorete, que sempre se esforçaram em conceder uma ótima educação para mim e meus irmãos. Mesmo nos momentos de tantas dificuldades, eles não desistiram, e por meio do exemplo de garra deles, eu mantive meu empenho para realizar o meu sonho profissional.

Não posso deixar de agradecer aos meus irmãos, Israel e Quelren. Que me acompanharam e torceram pela minha jornada.

Ao meu primo Adriano, conhecido como “bola”, por inspirar e incentivar não só a mim como também aos meus irmãos na conquista profissional.

Ao meu orientador Prof. Francisco Januário, pela disposição, confiança, conselhos, amizade, a mim significaram muito.

Ao técnico de laboratório, veterano e amigo, Arllem Farias, que desde o início da graduação, cedeu computador quando eu não tinha, topou ser supervisor do meu estágio, e pela amizade durante o meu curso.

Aos meus amigos de curso, Felipe Menezes e Josias Ben, que apoiaram e atuaram em todos os momentos durante esses anos de curso.

E a todos aqueles que contribuíram diretamente ou indiretamente na minha caminhada. Muito obrigado!!!

RESUMO

No fim do ano de 2019, o mundo vivenciou o surgimento de uma nova variante de vírus da família coronavírus, denominado COVID-19. Este vírus provocou distúrbios respiratórios em grande parte da população, e expôs as limitações do sistema de saúde, possibilitando uma oportunidade para o uso de tecnologias para o acompanhamento médico. Muitas pessoas ficaram internadas, e um dos critérios para esta ação, é a análise da oximetria dos pacientes. Esta monografia tem por objetivo propor um sistema de rotulação da gravidade de pacientes, através de um conjunto de sensores, provendo, por meio de um sistema web, os dados sobre batimentos cardíacos e oximetria, para a análise do profissional da saúde em qualquer parte do mundo. Com base nessa abordagem, o sistema faz uso de NodeMCU, de um sensor de batimentos cardíacos e oximetria MAX30100, Raspberry Pi 2 e um sistema web.

Palavras-chave: oximetria, COVID-19, NodeMCU, MAX30100, Raspberry Pi 2, sistema web.

ABSTRACT

At the end of 2019, the world experienced the emergence of a new variant of the coronavirus family virus, called COVID-19. This virus caused respiratory disorders in a large part of the population, and exposed the limitations of the health system, providing an opportunity for the use of technologies for medical monitoring. Many people were hospitalized, and one of the criteria for this action is the analysis of the patients' oximetry. This monograph aims to propose a system of labeling the severity of patients, through a web system, data on heartbeats and oximetry, for the analysis of the health professional in any part of the world.. Based on this approach, the system makes use of NodeMCU, a MAX30100 heart rate and oximetry sensor, Raspberry Pi 2 and a web system.

Keywords: oximetry, COVID-19, NodeMCU, MAX30100, Raspberry Pi 2, web system.

LISTAS DE FIGURAS

Figura 1 - Distribuição do sangue no sistema circulatório.....	17
Figura 2 - Etapas da formação da hemoglobina.....	18
Figura 3 - Diagrama de blocos do oxímetro.....	20
Figura 4 - Fotopletismografia de dedo.....	21
Figura 5 - Blocos de construção da IoT.....	23
Figura 6 - Arquitetura do sistema.....	27
Figura 7 - Fluxograma do projeto.....	28
Figura 8 - Diagrama de caso de uso.....	29
Figura 9 - Sensor MAX30100.....	30
Figura 10 - Pinagem do sensor MAX30100.....	31
Figura 11 - ESP8266 NodeMCU.....	32
Figura 12 - Raspberry Pi 2.....	33
Figura 13 - Montagem do circuito.....	34
Figura 14 - Esquema recomendado para uso da MAX30100.....	34
Figura 15 - Formato de mensagem enviada ao servidor.....	35
Figura 16 - Estrutura de diretórios do back-end.....	36
Figura 17 - Esquema para registros.....	38
Figura 18 - Esquema para sensores.....	39
Figura 19 - Fluxo de transição de tela.....	40
Figura 20 - Tela de sensores.....	40
Figura 21 - Tela de histórico do sensor.....	41
Figura 22 - Estrutura de diretórios do front-end.....	41
Figura 23 - Oxímetro MedMax AS-301-L.....	44
Figura 24 - Requisição para cadastro de sensor.....	45
Figura 25 - Requisição para listagem de sensores.....	45

Figura 26 - Requisição para histórico do sensor.....	46
Figura 27 - Oxímetro de pulso comercial e coleta de do oxímetro do sistema.....	46
Figura 28 - Resultado oxímetro de pulso do projeto.....	47
Figura 29 - Coletas simultâneas.....	49
Figura 30 - Resultado coletas simultâneas.....	49
Figura 31 - Cor leve dos resultados de rotulagem.....	50
Figura 32 - Cor moderada dos resultados de rotulagem.....	50
Figura 33 - Cor severa dos resultados de rotulagem.....	51
Figura 34 - Listagem de sensores.....	52
Figura 35 - Histórico de leituras.....	52

LISTAS DE TABELAS

Figura 1 - Resultado coletas.....	47
-----------------------------------	----

SUMÁRIO

INTRODUÇÃO.....	13
1.1 Objetivo Geral.....	15
1.2 Objetivos Específicos.....	15
1.3 Organização da Monografia.....	15
FUNDAMENTAÇÃO TEÓRICA.....	16
2.1 Sistema Circulatório.....	16
2.2 Hemoglobina.....	17
2.3 Determinação de oxigenação no sangue.....	18
2.4 Oxímetro.....	19
2.5 Fotopletismografia.....	20
2.6 Internet das coisas (IoT).....	21
2.7 Aplicação Web Cliente-Servidor.....	23
2.8 NodeJS.....	24
2.9 ReactJS.....	24
2.10 MongoDB.....	25
DESENVOLVIMENTO.....	26
3.1 Metodologia.....	26
3.2 Arquitetura do projeto.....	26
3.3 Fluxograma do projeto.....	27
3.4 Diagrama de caso de uso.....	29
3.5 MAX30100.....	29
3.6 ESP8266.....	31
3.7 Raspberry Pi 2.....	32
3.8 Sensoriamento de Oximetria.....	33

3.9 Processamento e rotulação.....	35
3.10 Projeto do Back-end.....	36
3.11 Banco de dados MongoDB.....	37
3.12 Projeto Front-end.....	39
3.13 Bibliotecas utilizadas.....	43
RESULTADOS.....	44
4.1 Execução de testes nas rotas do back-end.....	44
4.2 Leituras.....	46
4.3 Acurácia.....	47
4.4 Coletas simultâneas.....	48
4.5 Interface Sistema Web.....	49
CONCLUSÕES.....	53
5.1 Considerações Finais.....	53
5.2 Trabalho Futuro.....	53
REFERÊNCIAS BIBLIOGRÁFICAS.....	54

INTRODUÇÃO

Os sinais vitais apresentam uma grande importância para fornecer informações sobre o estado de saúde de um indivíduo, pois estes são os melhores indicadores das alterações ocorridas nos sistemas circulatório, respiratório, renal ou endócrino. Os sinais vitais são definidos como parâmetros do funcionamento regular dos órgãos vitais, sendo realizado as verificações e análises da pressão arterial, temperatura corporal, respiração e pulsação. (MURTA; *et al*, 2009, p.425).

Dadas as circunstâncias de obtenção de sinais cardíacos, atualmente, a oximetria tem notoriedade nos estudos de trocas gasosas do corpo humano e vem ganhando importância na detecção de doenças cardiológicas. Em específico, as cardiopatias congênitas compostas por malformação cardíaca e aspectos fisiológicos distintos, são as principais causadoras da morbidade neonatal, devido ao atraso no diagnóstico e tratamento, e que poderiam ser detectadas a partir da análise de oximetria. (MEDEIROS *et al*, 2015).

Atualmente, os oxímetros de pulso são instrumentos utilizados para a mensuração do nível de saturação de oxigênio no sangue arterial. São equipamentos que não necessitam de tomadas de ações invasivas para a obtenção de informações. Eles utilizam sensoramento óptico posicionados sobre a pele para que haja leitura da variação do sangue em uma determinada zona vascularizada do corpo humano.

Arelado à importância de obter conhecimentos sobre o quadro de oxigenação no corpo humano. Em 2019, o mundo se deparou com uma nova variante de vírus, da família coronavírus, denominado COVID-19. Foram descritos casos de pacientes com quadros de hipoxemia, quadro clínico na qual o paciente apresenta uma baixa concentração de oxigênio no sangue arterial, caracterizado como grave na ausência de dispneia, alegado como “hipoxemia silenciosa”. Foi recomendado para os pacientes de grupo de risco e com quadro não agravado, que monitorassem a saturação de oxigênio no sangue, de tal modo, que houvesse o acompanhamento médico se fosse notado o decaimento para um nível de oxidação abaixo de 90% (LUKS, SWENSON, 2020).

A situação sanitária atual demonstra que o acesso aos mecanismos de saúde podem ser comprometidos por eventos não previstos com antecedência, e que nesses cenários a população de menor renda fica mais exposta, principalmente por conta de

fatores econômicos, ocasionando o colapso do sistema de saúde. (PIRES; CARVALHO; XAVIER, 2020). Esta citação pode ser ratificada pelo portal Médicos sem Fronteiras, que relatou o caso ocorrido na cidade de Manaus, onde em Janeiro de 2021, a cidade vivenciou um colapso no sistema de saúde por causa da falta de oxigênio, fator essencial para o tratamento de casos graves da COVID-19. (PALHA, 2021).

A telemedicina é bastante conhecida pelos profissionais da área da saúde. Há o uso de tecnologias de telecomunicações, com o propósito de fornecer informações e serviços médicos aos profissionais. É possível realizar consultas remotas, ou até mesmo, a realização de exames para obter uma avaliação médica (PEREDNIA, ALLEN, 1995).

Atualmente, com o avanço tecnológico de microprocessadores e sensores com alto desempenho, é notório a possibilidade de refinar o conceito de telemedicina com o uso dessas novas tecnologias. Sendo possível promover uma facilitação de disposição de dados vitais, para os profissionais, agregando atendimento ágil e priorizado para os pacientes de quadro de risco. Utilizar uma cadeia de sensores conectados à internet das coisas, acarreta o fornecimento de dados vitais em tempo real para o responsável médico.

Dessa forma, a solução apresentada nesta monografia envolve o desenvolvimento de um sistema IoT completo, composto por: sensores de oximetria, microcontroladores, e um sistema web para acompanhamento em tempo real dos dados vitais fornecidos pelos sensores. O sistema se inicia com os sensores de oximetria MAX30100 conectados ao microcontrolador NodeMCU ESP8266, na qual este possui uma lógica de programação para a calibragem e coleta de informações por meio dos sensores. O NodeMCU, após obter os dados, enviará essas informações para o Raspberry Pi 2, onde será processado e realizado a rotulagem das informações dos dados vitais, e posteriormente enviará as rotulagens para o servidor do sistema web, proporcionando a visualização do profissional da saúde em qualquer lugar que ele esteja.

Sendo assim, acredita-se que o desenvolvimento da aplicação deste sistema, possibilitará aos diversos profissionais a melhor análise e controle para tomadas de decisões relacionadas ao quadro clínico de cada paciente.

1.1 Objetivo Geral

Propor um sistema de coleta de oximetria simultânea de pacientes e disponibilizar os dados adquiridos em um sistema web para consulta de profissionais da saúde, por meio de sensores MAX30100, ESP8266 NodeMCU e um Raspberry Pi 2.

1.2 Objetivos Específicos

- a. Propor uma arquitetura de sistema de hardware e software a partir do levantamento de requisitos baseados nas necessidades de coleta de dados de oximetria em tempo real.
- b. Desenvolver um sistema para coleta de dados de oximetria e a transmissão de dados via protocolo MQTT.
- c. Desenvolver um sistema para processamento e rotulação de dados de oximetria e transmitir estes para um servidor em nuvem via protocolo HTTP.
- d. Desenvolver um sistema web apresentando informações de oximetria do paciente em tempo real e histórico das últimas coletas.
- e. Realização de testes de coleta de sensoriamento simultâneos.

1.3 Organização da Monografia

A Monografia está organizada da seguinte maneira:

- No Capítulo 2, são apresentados os fundamentos teóricos, que darão contexto para o desenvolvimento deste trabalho.
- No Capítulo 3, são demonstradas as etapas do desenvolvimento do sistema, informando em detalhes os componentes físicos utilizados, frameworks para desenvolvimento de software, bibliotecas utilizadas e arquitetura em si do projeto.
- No Capítulo 4, são comentados os resultados obtidos no trabalho, com um olhar relacionado a interface de usuário, também como uma análise comparativa com um oxímetro de pulso comercial, sendo assim, é verificada a precisão da coleta de dados do sistema.
- No Capítulo 5, são descritos as conclusões acerca do trabalho desenvolvido e a proposta de otimização para trabalhos futuros.

2. FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão abordados os conceitos ligados ao contexto da área da saúde e das tecnologias utilizadas para o desenvolvimento deste trabalho.

2.1 Sistema Circulatório

O sistema circulatório tem como propósitos, manter o organismo saudável e garantir a sobrevivência das células, por meio de funções, tais como, o transporte de nutrientes, eliminação de agentes metabólicos, transporte de hormônios, transporte de oxigênio, entre outras. (HALL; GUYTON, 2011).

O sistema circulatório possui coração e vasos sanguíneos controlados de tal forma que produz débito cardíaco e pressão arterial necessária, acarretando um fluxo sanguíneo suficiente para atender todas as necessidades do organismo. A circulação pode ser dividida entre circulação sistêmica e circulação pulmonar, na qual a circulação sistêmica irriga os tecidos corporais e órgãos. E a circulação pulmonar promove a difusão do oxigênio, presente nos alvéolos pulmonares, para dentro do sangue, e no sentido oposto, promove a difusão de dióxido de carbono, para fora do sangue. Esta ação ocorre devido a capacidade da hemoglobina em se combinar com o oxigênio dos pulmões, e posteriormente realiza a liberação deste oxigênio nos capilares teciduais periféricos. (HALL; GUYTON, 2011). A Figura 1, exibe a distribuição do sangue no sistema circulatório.

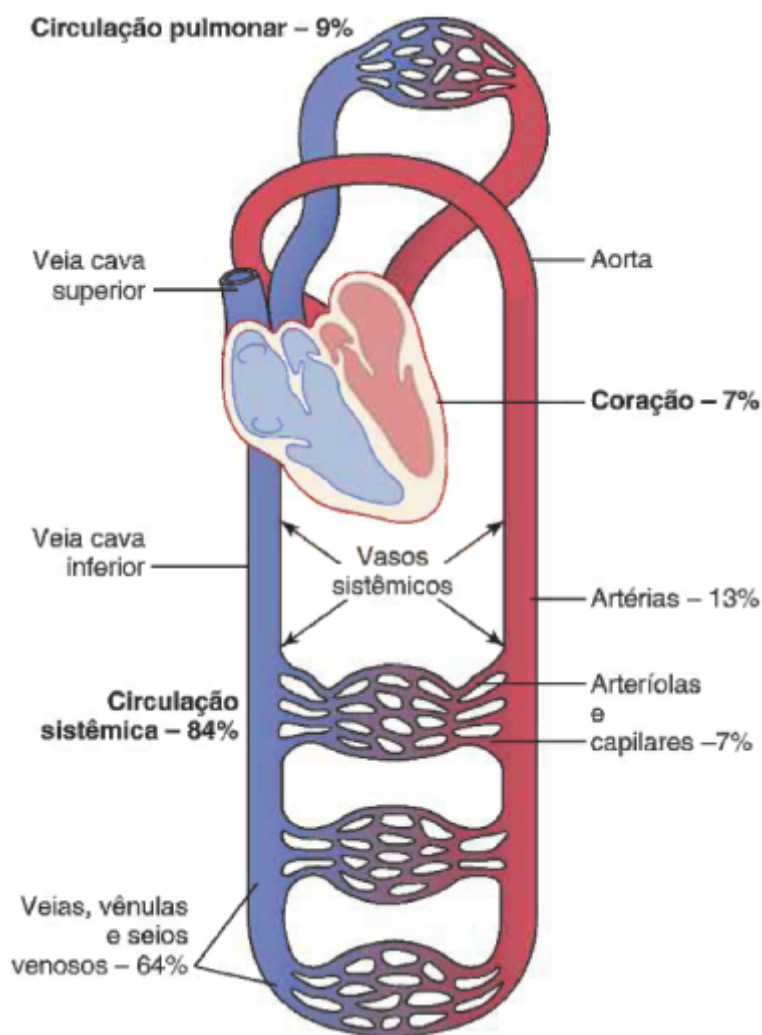


Figura 1 - Distribuição do sangue no sistema circulatório.

Fonte: (HALL; GUYTON, 2011).

2.2 Hemoglobina

A hemoglobina é uma proteína presente nos glóbulos vermelhos, sendo responsável pela coloração vermelha do sangue, e possui o transporte de oxigênio como sua característica principal. A sua formação química é dividida em quatro etapas. A primeira com a ligação entre a succinil-coA com a glicina formando o pirrol. Quatro moléculas de pirrol se combinam para a formação da protoporfirina IX, que combina com o ferro para formar a molécula heme. Cada molécula heme é combinada com a cadeia polipeptídica globina, formando a subunidade da hemoglobina referida como cadeia de hemoglobina. (HALL; GUYTON, 2011). A Figura 2 exibe as etapas desse processo.

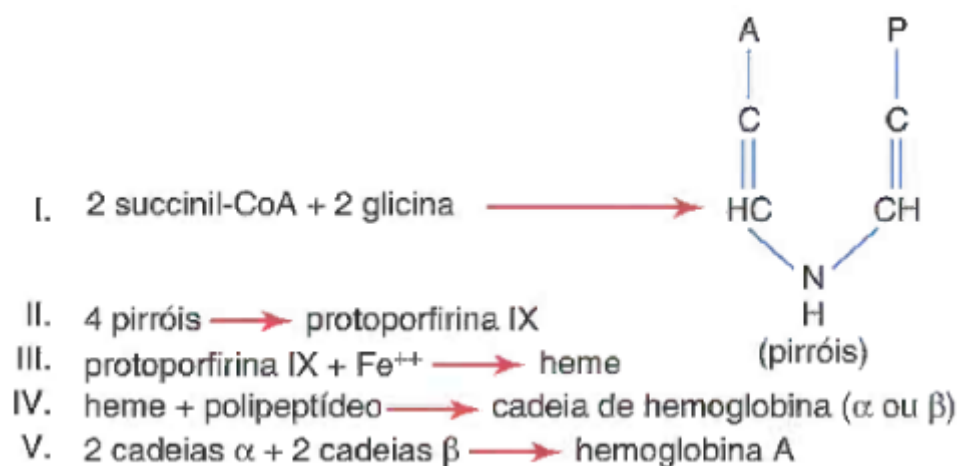


Figura 2 - Etapas da formação da hemoglobina.

Fonte: (HALL; GUYTON, 2011).

2.3 Determinação de oxigenação no sangue

Há diversas formas para se mensurar a saturação de oxi-hemoglobina no sangue, podendo ser por métodos químicos ou ópticos, invasivos ou não invasivos. Nos métodos ópticos possuem destaque a espectrofotometria que é a base de todos os oxímetros. A ideia consiste em um experimento em vidro, na qual se utiliza um fotodetector para converter a intensidade luminosa incidente de uma determinada substância (I_0) e a intensidade luminosa transmitida através dessa substância (I) em corrente elétrica. Sendo assim, é possível realizar a transmitância (T) padrão da substância. (MEHRABI, 2018). A Equação 1 apresenta o cálculo da transmitância.

$$T = I/I_0 \quad (1)$$

A partir da transmitância, é possível determinar o coeficiente de absorção (A). A Equação 2 apresenta o cálculo do coeficiente de absorção.

$$A = 2 - \log(T) \quad (2)$$

A partir do coeficiente de absorção definido, é possível encontrar a quantidade da substância dentro de uma solução através da lei de Beer. (MEHRABI, 2018). A Equação 3 demonstra o cálculo da lei de Beer.

$$A = \epsilon(\lambda)Csd \quad (3)$$

Onde $\epsilon(\lambda)$ é o coeficiente de extinção da substância dado um comprimento de onda λ , d é o comprimento do trajeto de luz e Cs é a concentração da substância.

De acordo com Garzón, a saturação do sangue encontrada pode ser definida por quatro parâmetros, onde hipoxia significa a diminuição a baixa concentração de oxigênio no sangue (Garzón, *et al*, 2022), são eles:

- Normal: entre 95% e 99% de saturação;
- Hipoxia leve: entre 91% e 94% de saturação;
- Hipoxia moderada: entre 80% e 90% de saturação;
- Hipoxia severa: abaixo de 80% de saturação;

2.4 Oxímetro

Os oxímetros de pulso são utilizados para determinar o nível de saturação de oxigênio (SpO_2) no sangue de forma não invasiva. Neste equipamento, são utilizados sensores ópticos posicionados sobre a pele, de tal modo que possa emitir e recepcionar dois comprimentos de onda distintos, espectro vermelho e infravermelho. (CALIL, *et al*. 2002).

Na grande maioria dos equipamentos, consiste em uma unidade de processamento de dados com visor informando os valores coletados, referentes aos sinais emitidos e recepcionados pelo sensor. Uma característica comum nos oxímetros, é a indicação da frequência cardíaca, que pode ser obtida a partir da análise do sinal pletismográfico. (CALIL, *et al*. 2002). A Figura 3 representa o diagrama de blocos de um oxímetro.

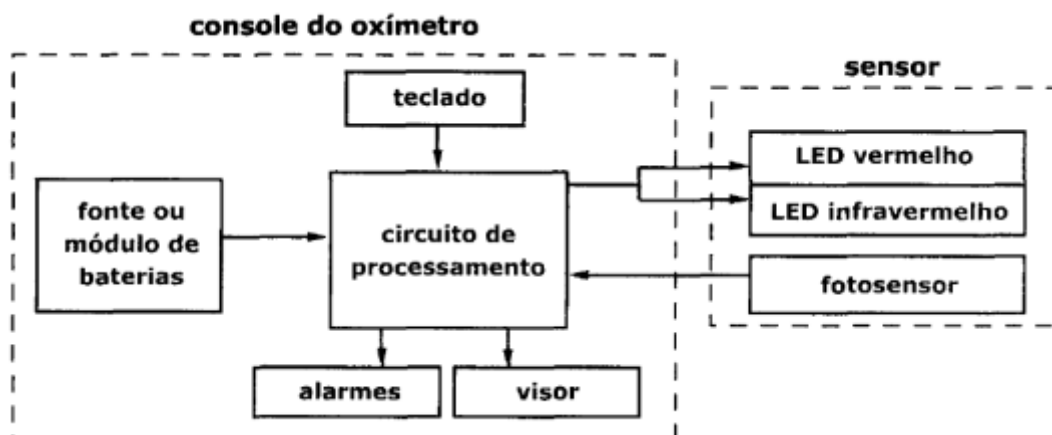


Figura 3 - Diagrama de blocos do oxímetro.

Fonte: (CALIL, et al. 2002).

A espectrofotometria sanguínea é o princípio, na qual é feita a obtenção do sinal da oximetria de pulso. Este método consiste na medição da quantidade de luz transmitida ou refletida através das veias capilares dos pacientes. Em razão dos diferentes níveis de oxigenação, o sangue rico em oxihemoglobina apresenta uma coloração mais vermelha e brilhante, diferente do sangue com baixo nível de oxigênio, desoxihemoglobina, que apresenta uma coloração mais azulada. (CALIL, et al. 2002).

2.5 Fotopletismografia

A fotopletismografia tem como objetivo emitir raios infravermelhos provocando reflexos nos glóbulos vermelhos presentes nas veias dos pacientes. Os resultados são transmitidos para um computador, possibilitando a visualização para o profissional da saúde. Para que este sinal de fotopletismografia seja interpretado adequadamente pelos profissionais, é realizada uma filtragem do sinal, uma vez que o circuito de aquisição do sinal, pode sofrer interferências em outras frequências, acarretando a distorção do sinal. As causas são (ELGENDI, 2012):

- Interferência por rede elétrica: normalmente ocorre entre 50 e 60 Hz. É causada por equipamentos que consomem energia de corrente alternada da rede elétrica;
- Interferência motora: causada pelo movimento dos sensores enquanto ocorre a coleta do sinal. É comum essa interferência na fotopletismografia de dedo e pulso, principalmente quando essa aquisição ocorre durante o acompanhamento de atividades físicas. Possui baixas frequências, causando tendências no sinal.

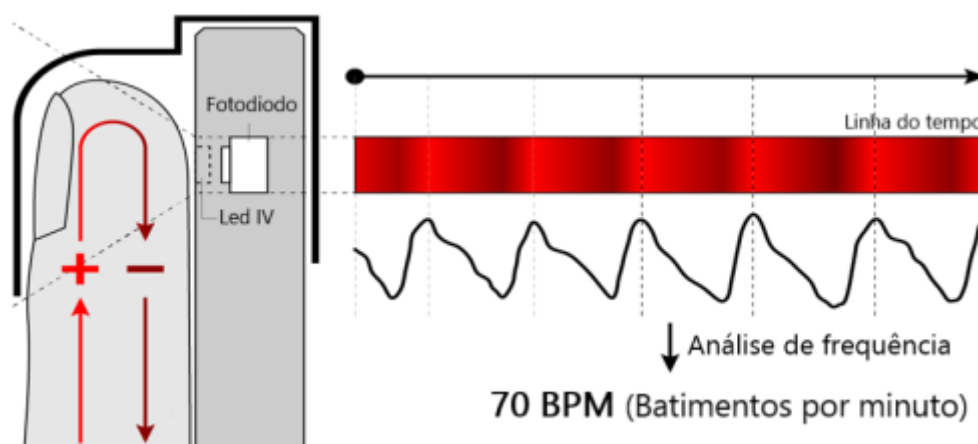


Figura 4 - Fotopletismografia de dedo.

Fonte: (BARCH DAILY, 2014).

A Figura 4 apresenta o sinal de fotopletismografia em uma aquisição feita na ponta do dedo.

2.6 Internet das coisas (IoT)

A internet das coisas (IoT) surgiu a partir dos avanços tecnológicos nas áreas da eletrônica, como sistemas embarcados, microeletrônica e instrumentação. Esta tecnologia utiliza a robustez da internet atual, com o propósito de integrar objetos diversos do dia-a-dia com o objetivo de viabilizar tanto o controle remoto quanto permitir a visualização de dados dos dispositivos em qualquer distância. Isto dispõe de aplicações em tempo real, tais como, a comunicação a distância entre usuários de produtos, monitoramento remoto de pacientes ou até mesmo o sensoriamento de processos fabris (SANTOS, *et al.* 2016).

Existem limitações de processamento, memória, conectividade e energia. Os novos desafios surgem no decorrer dos desenvolvimentos de novas aplicações de IoT para prover dados que serão objetos de estudos para as futuras aplicações e para a adaptação de protocolos de rede. O objetivo é que esta tecnologia seja capaz de suprir as expectativas dos usuários (SANTOS, *et al.* 2016).

Essa tecnologia pode ser definida em 6 blocos de construção, são eles. (SANTOS, *et al.* 2016):

- Identificação: Tem como propósito identificar os objetos para assim poder conectá-los à internet. São utilizadas tecnologias como RFID, NFC e endereçamento de IP.
- Sensores ou atuadores: Tem como propósito coletar dados ou executar ações nos objetos. Os dados coletados são encaminhados para plataformas de controle ou armazenamento, tais como clouds e data warehouse. E a execução se realiza a partir de comandos provenientes das plataformas de controle.
- Comunicação: Refere-se ao tipo de conexão entre os objetos e a internet. Geralmente são adotadas algumas opções de tecnologias como Wi-Fi e Bluetooth.
- Computação: É a unidade de processamento de dados como microcontroladores responsáveis pelas execuções de algoritmos.
- Serviços: Refere-se ao tipo de coleta, processamento e atuação que IoT está inserida, são elas:
 - Identificação: Mapeia as entidades físicas ou virtuais, como exemplo, a temperatura de um processo.
 - Agregação de dados: Coletam e processam as informações obtidas dos objetos.
 - Colaboração e inteligência: Agregam os dados para tomadas de decisões baseadas em configurações pré estabelecidas.
 - Ubiquidade: Promovem serviços de colaboração e inteligência a qualquer momento.
- Semântica: Trata-se do uso eficiente dos recursos da IoT a partir de dados existentes. São utilizadas técnicas como *Resource Description Framework* (RDF), *Web Ontology Language* (OWL) e *Efficient XML Interchange* (EXI).



Figura 5 - Blocos de construção da IoT.

Fonte: (SANTOS, *et al.* 2016).

A Figura 5, exibe os blocos de construção ligados à tecnologia de internet das coisas.

2.7 Aplicação Web Cliente-Servidor

O software é um produto que se faz presente no cotidiano das pessoas, seja ele acessado por um computador, por smartphone, automóveis e entre muitos outros dispositivos. Ele não é uma máquina, e sim uma descrição de máquina, em outras palavras, o software é incapaz de realizar trabalho ao menos que exista uma máquina que interprete as informações contidas nele, resultando uma interação com o usuário. (FERNANDES, 2003).

No âmbito do desenvolvimento de software, encontra-se o desenvolvimento web, que são aplicações de uma solução executadas diretamente por meio de um browser (ou navegador). O desenvolvimento web segue 3 padrões, de acordo com o portal da UFPE, são eles:

- Cliente web thin: São aplicativos com base na internet, em que há pouco controle de configuração do cliente. Toda a lógica de negócio é gerenciada por um servidor.
- Cliente Thick web: A lógica de negócios é executada diretamente na máquina do cliente. De modo geral, o cliente utiliza HTML dinâmico, Applets Java ou controles ActiveX para a execução da lógica. A comunicação com o servidor ainda é realizada por protocolos HTTP.
- Entrega pela web: Além do uso de protocolos HTTP, a comunicação do cliente com o servidor, também é realizada por meio de outros protocolos, tais como IIOP e DCOM, protocolos estes utilizados para suportar sistemas de objetos distribuídos.

No desenvolvimento cliente-servidor, podem ser utilizadas diversos frameworks e bancos de dados, podemos citar o NodeJS, empregado na maioria das vezes nos servidores, o ReactJS, empregado no cliente e o banco de dados não relacional MongoDB.

2.8 NodeJS

O NodeJS é um framework que permite a interpretação da linguagem de programação Javascript/Typescript de forma assíncrona, seu código é aberto e é orientado a eventos. Este framework foi criado por Ryan Dahl em 2009, com o propósito de portar a programação Javascript para ser utilizada nos servidores. Ele permite o desenvolvimento de aplicações escaláveis em rede, e a cada requisição recebida, é criada uma thread para tratá-la, demandando recursos como memória RAM. Quando estes recursos atingem o limite, as novas requisições serão postas em espera até ocorrer liberação de memória. (OPUS SOFTWARE, 2020).

2.9 ReactJS

O ReactJS é um framework, que assim como o NodeJS, ele também permite a interpretação das linguagens de programação Javascript/Typescript, criado pelo Facebook em 2011 para desenvolvimentos de interfaces de usuário na web. Este framework veio com objetivo de otimizar as atualizações e as sincronizações de

atividades simultâneas. Ele possui conexão simplificada entre HTML, CSS e Javascript/Typescript. Seu maior objetivo é conectar diferentes partes de uma página, logo seu funcionamento é administrado por componentes. (KENZIE, 2020).

2.10 MongoDB

MongoDB é um banco de dados não relacional NoSQL, de alta performance e flexível. Os bancos NoSQL apresentam vantagens comparados a outros, quando a necessidade é escalabilidade, flexibilidade, bom desempenho e facilidade para consultas. Ele é orientado a documentos, em outras palavras, os dados são armazenados como documentos ao invés de serem armazenados em tabelas, como é realizado com os bancos relacionais SQL. Este banco é muito utilizado para ambientes com poucos recursos, onde possibilita o suporte de qualquer servidor para atender o usuário. Na figura 10, é exibida a integração do banco de dados com os demais componentes do sistema web. (TREINAWEB, 2021).

3. DESENVOLVIMENTO

Neste capítulo será descrito o desenvolvimento da solução proposta, tendo como objetivo analisar, rotular e exibir os dados vitais de saturação de oxigênio no sangue (SpO_2) e batimentos cardíacos, adquiridos por sensoramento de oximetria, em um sistema web.

3.1 Metodologia

Para atender a lista de objetivos específicos propostos na seção 1.2, o desenvolvimento do projeto se deu pelas seguintes etapas:

- Elaboração da arquitetura do projeto.
- Elaboração do fluxograma do projeto para serem avaliadas as ações do sistema.
- Elaboração do diagrama de caso de uso.
- Analisar e adquirir os componentes MAX30100, ESP8266 NodeMCU e Raspberry Pi 2.
- Desenvolvimento do dispositivo de sensoramento de oximetria.
- Desenvolvimento do sistema de processamento e rotulação de dados provenientes da coleta do sensor.
- Desenvolvimento do backend e frontend do sistema web.
- Integrar todos os componentes do sistema, do hardware até o software.

3.2 Arquitetura do projeto

O sistema pode ser dividido modularmente em 4 partes, o sensoramento de oximetria, processamento e rotulação de dados, aplicação web e backend para o qual a aplicação do sistema irá cadastrar e consultar dados de oximetria. Os módulos funcionam da seguinte forma: o usuário, paciente, através do sensor MAX30100, irá pôr o dedo no sensor para a leitura de dados, após um período de 10s, o dispositivo enviará as informações para o Raspberry Pi 2, através do protocolo MQTT, na qual em sua programação há o algoritmo de processamento e rotulação dos dados de saturação do oxigênio (SpO_2) e frequência cardíaca. A partir dos dados já processados, estes serão enviados ao backend por meio do protocolo HTTP, onde será armazenado as

informações obtidas em um banco de dados não relacional, e assim, poderá ser consultado através da interface de usuário acessada pelo usuário, profissional da saúde. A Figura 6 demonstra a arquitetura do sistema.

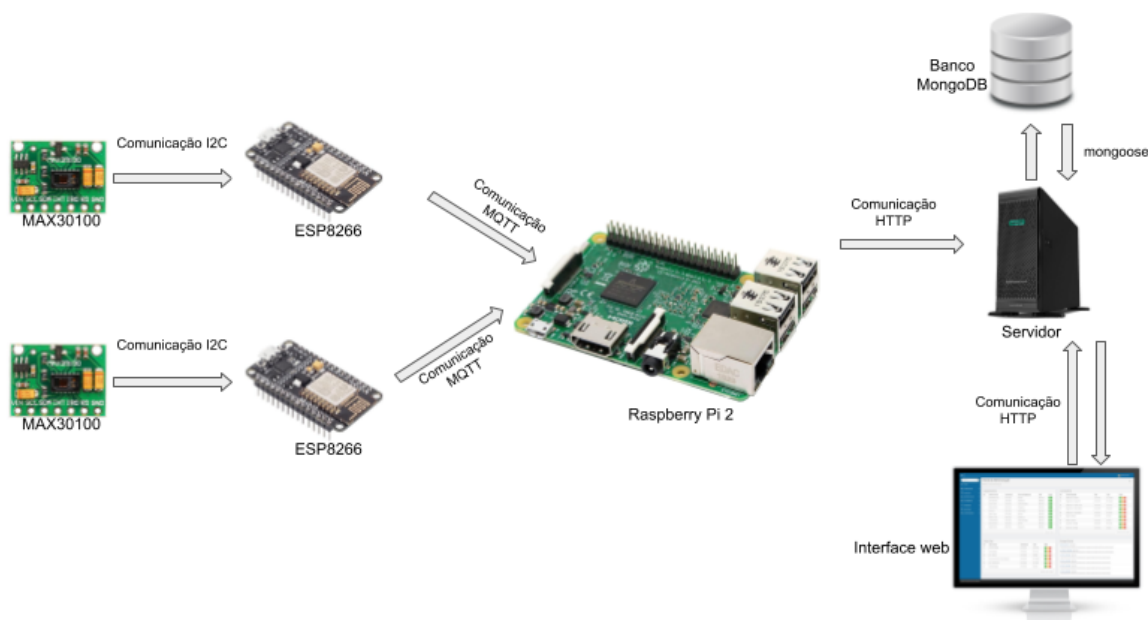


Figura 6 - Arquitetura do sistema

Fonte: Próprio autor

3.3 Fluxograma do projeto

O fluxograma do projeto foi construído para mapear graficamente as etapas de algoritmos empregados dentro do ESP8266, do Raspberry Pi 2, back-end e front-end, evidenciando a sequência lógica de ações, respostas e o fluxo de dados. A Figura 7 representa o fluxograma do projeto com o propósito de identificar os eventos do sistema.

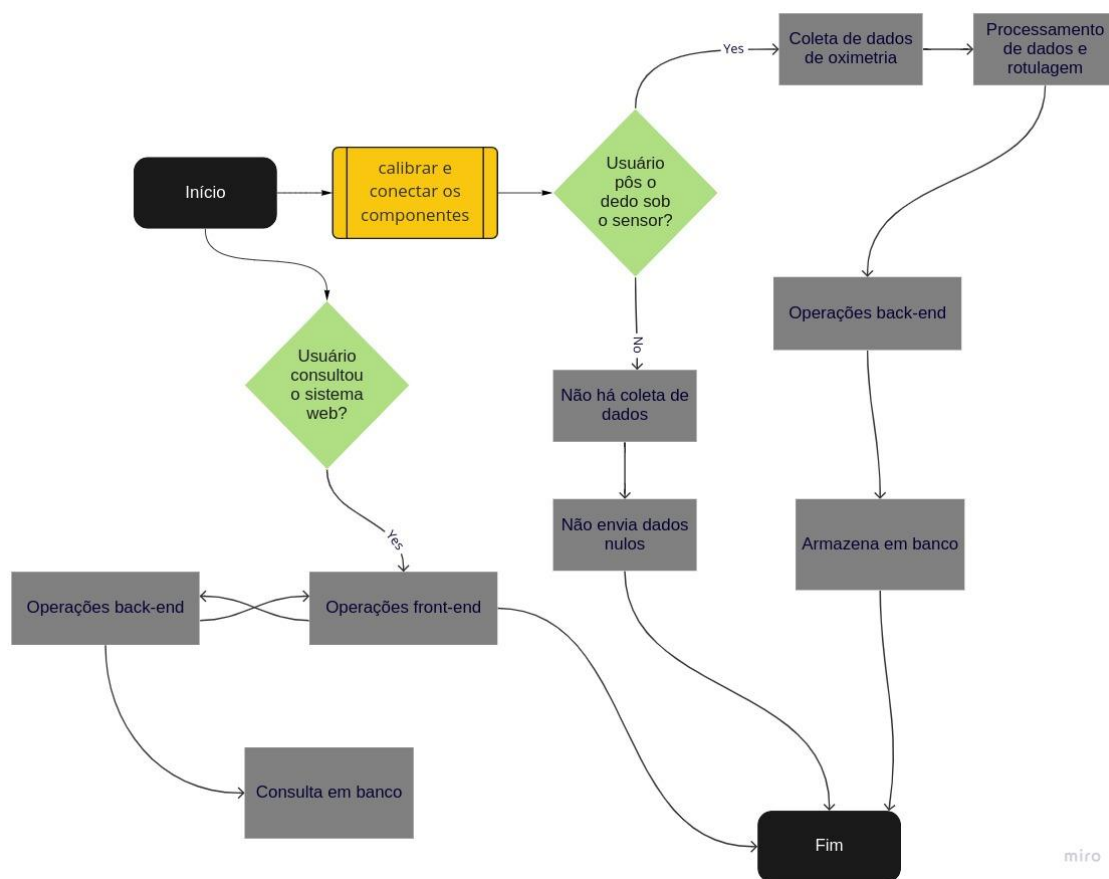


Figura 7 - Fluxograma do projeto.

Fonte: Próprio autor.

No início o projeto possui duas ramificações, a ramificação amarela, mostra a ação pré-definida, que é a calibração dos componentes de hardware e a conexão dos mesmos à rede. A ramificação triangular verde partindo do início é a ação do usuário, profissional da saúde, em relação ao sistema web, caso o usuário tenha contato, será realizado as operações do front-end, como renderização de dados já cadastrados no sistema. A ramificação triangular ligada ao bloco amarelo, representa as ações de usuários, pacientes, de acordo com sua interação com o sensor, caso não haja interação, a leitura nula não será enviada para armazenagem no servidor, caso haja leitura, os dados do sensor seguirão para o processamento e rotulagem, sendo assim direcionados para as operações no back-end, que envolve o cadastro dessas informações no banco de dados do sistema.

3.4 Diagrama de caso de uso

O diagrama de caso de uso foi o recurso adotado no projeto para modelar o comportamento do sistema em relação às ações do usuário. No sistema haverá dois tipos de usuários, o paciente e o profissional da saúde. O paciente desempenha um único papel, que é fornecer os seus dados vitais de oximetria através do sensor. O profissional da saúde poderá fazer dois tipos de consultas, a consulta do resultado de cada sensor e acessar o histórico de registros de um determinado sensor. A Figura 8 representa o mapeamento realizado pelo diagrama de caso de uso.



Figura 8 - Diagrama de caso de uso.

Fonte: Próprio autor.

3.5 MAX30100

Para o projeto foi utilizado o módulo MAX30100, que é o módulo utilizado na implementação da solução deste trabalho. Ele conta com o uso de dois LEDs, um com comprimento de onda de 660 nm (espectro vermelho) e outro com 880 nm (infravermelho), conta também, com um fotodetector, um otimizador ótico e um circuito analógico de baixo ruído para a realização de processamento de sinais de fotopletismografia a fim de se obter as informações de oximetria e frequência cardíaca. Ele opera em baixas potências e possui um subsistema de medição de saturação de oxigênio (SpO_2) composto por um sistema de cancelamento de luz ambiente seguido

por um conversor analógico digital de 16 bits, cujo a amostragem do sinal, pode ser ser ajustado entre 50 e 1000 amostras por segundo. (MAX30100 Datasheet, 2014).

Além dos aspectos citados, o MAX30100, possui internamente um sensor de temperatura com o propósito de calibração do sistema de oximetria, podendo este ser submetido a variações decorrentes de temperatura ao operar em temperatura ambiente. (MAX30100 Datasheet, 2014). A Figura 9 mostra o design do sensor.



Figura 9 - Sensor MAX30100.

Fonte: (LABORATÓRIO DE GARAGEM, 2019).

O sensor é composto por 14 pinos, sendo disponíveis apenas 10 portas úteis e as demais utilizadas para a fixação da placa. Das 10 portas, possui-se duas para a comunicação I2C, sendo uma bidirecional e outra para entrada de clock, duas para controle de drivers dos LEDs, uma para alimentação dos drivers, duas para os LEDs, uma para os sistemas a uma para o aterramento. (MAX30100 Datasheet, 2014). A Figura 10 demonstra como é distribuído os pinos do sensor.

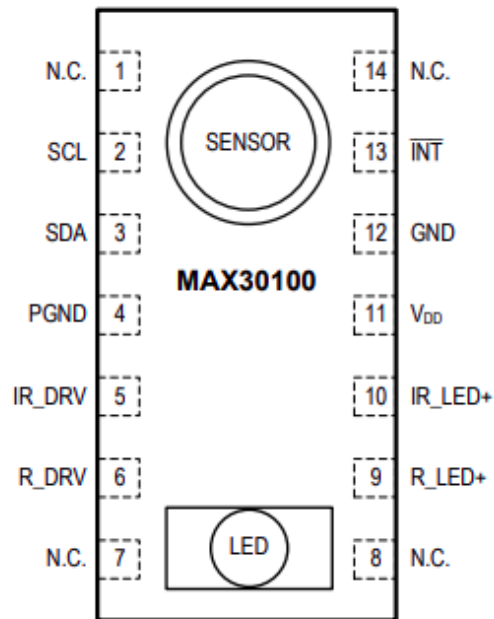


Figura 10 - Pinagem do sensor MAX30100.

Fonte: (MAX30100 Datasheet, 2014).

Uma vantagem do sensor é o fato de ele possuir biblioteca (*MAX30100_PulseOximeter.h*) para a programação na IDE do Arduino, sendo assim, é fácil a manipulação em conjunto com os demais periféricos do projeto.

3.6 ESP8266

Para ajudar no manuseio do sensor MAX30100, foi utilizado o microcontrolador NodeMCU ESP8266, que possui em sua estrutura uma solução WiFi SoC altamente integrativa aliado ao design compacto. A placa possui diversas aplicações no ramo de IoT. Ela conta com o chip ESP8266EX e microprocessador Tensilica Xtensa 32-bit LX106 RISC, que suporta a operação de sistema em tempo real e opera com clock ajustável de 80 MHz a 160 MHz. Possui também 128Kb de memória RAM e 4Mb de memória de armazenamento para dados e programas, e suporta as tecnologias de comunicação UART, SPI e I2C. (ESP8266 Datasheet, 2020). A Figura 11 mostra a placa ESP8266 utilizada no projeto

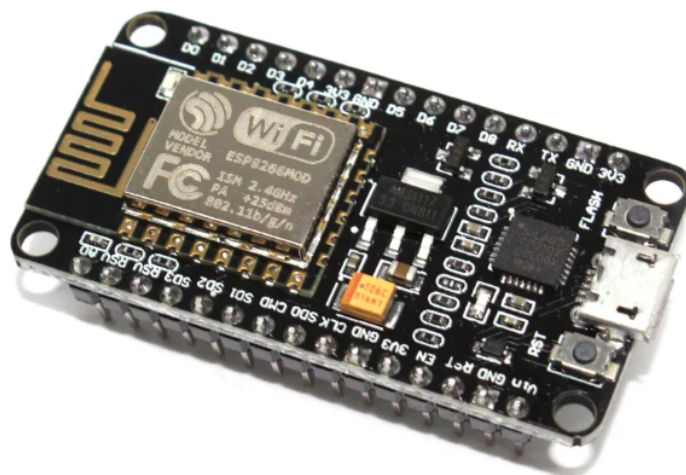


Figura 11 - ESP8266 NodeMCU.

Fonte: PISCALED.

Esta placa opera na tensão de 3.3V e aceita tensões de entrada na faixa de 7 a 12 V. Possui 16 pinos digitais I/O e um pino de entrada analógico (ADC). Seu WiFi é implementado utilizando as tecnologias TCP/IP e full 802.11 b/g/n WLAN MAC protocol. Ela permite uma programação através da IDE do arduino, esta IDE viabiliza o acesso a diversas bibliotecas, que aumentam a funcionalidade dos programas, podendo ser importados internamente da própria IDE, como também criar as próprias bibliotecas. Antes de operar a ESP8266, é necessário importar sua biblioteca (*ESP8266WiFi.h*). (ESP8266 Datasheet, 2020).

3.7 Raspberry Pi 2

Foi adotado para o projeto o Raspberry Pi 2, que é a segunda geração do Raspberry Pi, ele é um microcomputador, que desempenha papéis semelhantes a um computador, diferindo por ser compacto e possuindo todas as características de um computador em uma placa. Ele possui um processador quad-core ARM Cortex A7 rodando a 900MHz e 1GB de memória RAM, uma grande diferença dele para as versões atuais, é que ele não possui o chip Wi-Fi, sua conexão com a rede só consegue ser realizada via cabo. (RASPBERRY PI, 2015). A Figura 12 mostra a placa Raspberry Pi2 utilizada no projeto.

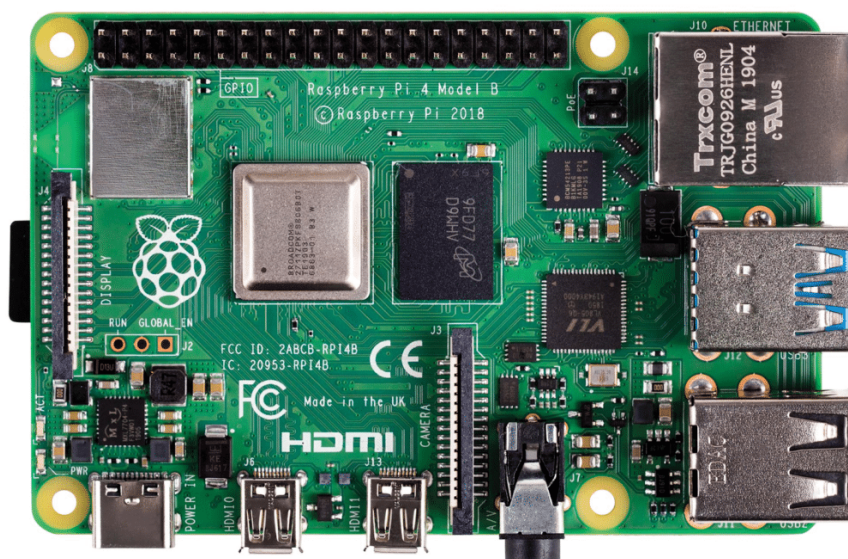


Figura 12 - Raspberry Pi 2.

Fonte: (EMBARCADOS, 2015).

Assim como um computador, o Raspberry pode atuar como servidor, hospedando diversas aplicações. O Raspberry também pode ser utilizado no desenvolvimento de serviços em python, com análises de dados e aplicações de *machine learning*.

3.8 Sensoriamento de Oximetria

Para o desenvolvimento do sensoriamento de oximetria, foram criados dois módulos de sensoriamento, cada um composto por um sensor MAX30100 e um ESP8266 NodeMCU. Esses módulos foram montados em uma protoboard, com o propósito de analisar e coletar as informações obtidas pelo monitoramento de dois sensores simultâneos. A Figura 13 demonstra a montagem do circuito montado.

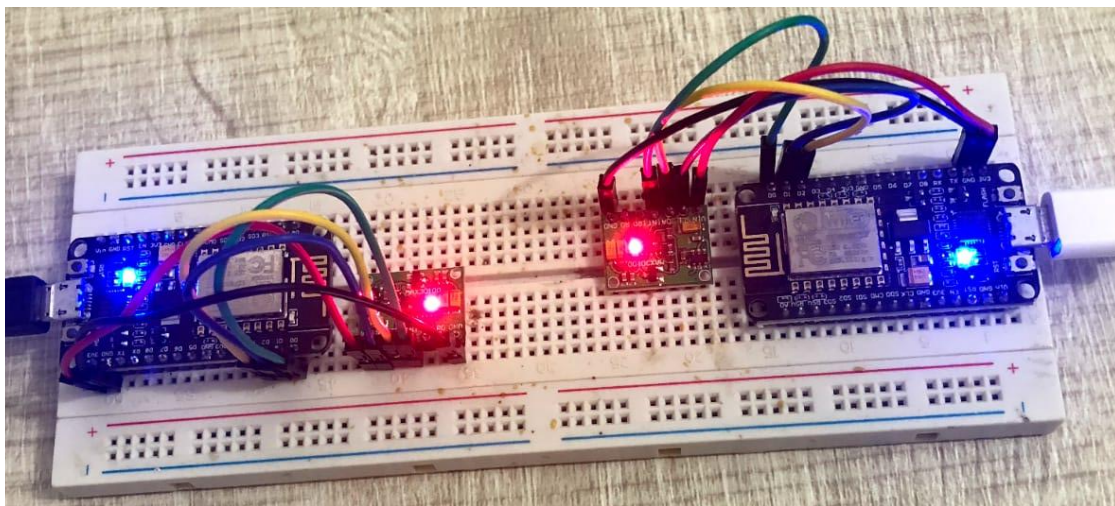


Figura 13 - Montagem do circuito.

Fonte: Próprio autor.

Essa montagem seguiu de acordo com a documentação do MAX30100 datasheet e de trabalhos já desenvolvidos com o uso do sensor MAX30100 e ESP8266 na área. A Figura 14 mostra o esquema do circuito recomendado para o uso da MAX30100.

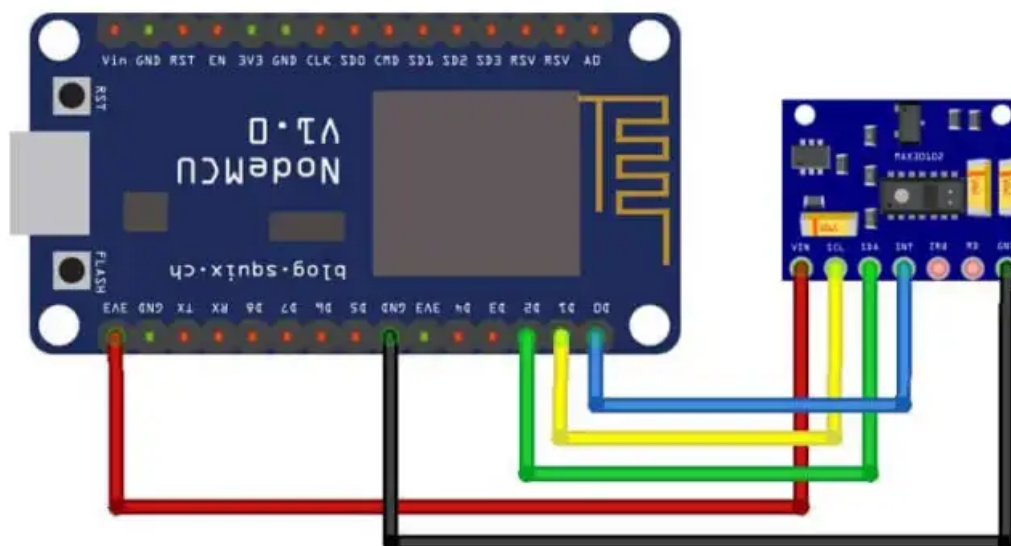


Figura 14 - Esquema recomendado para uso da MAX30100.

Fonte: (THE IOT PROJECTS, 2021).

No ESP8266, foi empregado um período de coleta de 10 segundos devido ao fato dos sensores estabilizarem suas leituras, fornecendo assim, resultados mais precisos da coleta. Essas informações são estruturadas em formato JSON (Javascript Object Notation), utilizando a biblioteca “*ArduinoJson.h*”. Os dados serão enviados, através de uma *publisher*, para o broken MQTT que se encontra no Raspberry, emitindo a mensagem através da flag “*/esp8266/data-collect*”, essa flag na programação é denominada de *topic*.

3.9 Processamento e rotulação

O processamento e rotulação dos dados são realizados no Raspberry. O Raspberry atua na intermediação do sistema, pois devido a sua capacidade computacional, ele viabiliza as análises inteligentes com os dados adquiridos. Neste projeto o Raspberry possui um serviço desenvolvido em python, que opera como *subscriber* no broken MQTT. Esse serviço fica conectado ao broken escutando todas as *publisher* que são emitidas para a flag “*/esp8266/data-collect*”, e é por meio da *publisher* que é possível obter a mensagem. Após receber as informações, este microcomputador realiza as rotulagens de oximetria, de acordo com os valores classificatórios mencionados no item 2.3 desta monografia, e assim envia os dados para o servidor da aplicação web, através do protocolo HTTP. A Figura 15 mostra o formato da mensagem enviada para o servidor.



```
{
  "sensorName": "ESP2",
  "oximetry": 92.0,
  "heartRate": 67,
  "label": "HIPOXIA LEVE"
}
```

Figura 15 - Formato de mensagem enviada ao servidor.

Fonte: Próprio autor.

3.10 Projeto do Back-end

Para o projeto do back-end que recebe as requisições tanto da aplicação web como do Raspberry, foi utilizado o framework NodeJS para o desenvolvimento. Uma estrutura de diretórios foi planejada a fim de organizar os arquivos e códigos fontes do projeto de forma componentizada, como é demonstrado na Figura 16.



Figura 16 - Estrutura de diretórios do back-end.

Fonte: Próprio autor.

A seguir são explanadas as funções de cada diretório e arquivos, pelos quais o back-end é composto, são eles:

- index.js: arquivo principal de cada diretório, utilizado como boa prática de programação para os imports. No caso do arquivo fora de todos os diretórios, é nele que é executado primeiro na programação, pois nele envolve as importações dos demais arquivos do sistema.
- database: neste diretório foram colocados os arquivos e diretórios responsáveis pela conexão e modelo do banco de dados.

- schemas: neste diretório se encontra os schemas do banco, que no caso foram criados dois, o esquema para o histórico de registros e o esquema para sensores.
- index.js: realiza a conexão com o banco de dados e relaciona os esquemas.
- middleware: neste diretório foram colocado as implementações de tratamento de erros dentro do sistema, evitando a parada do back-end para erros não mapeados.
- routes: neste diretório se encontra as rotas de requisições do sistema.
 - constraints: atua como validadores de dados de requisição do sistema, ou seja, antes de tratar a tarefa da requisição, é tratada primeiramente a informação da mensagem, ele valida se a estrutura da mensagem contida corresponde ao que o serviço da rota espera.
 - register: possui todas as rotas do sistema, POST para cadastros, e GET para consultas de dados.

Uma vez definida a estrutura de diretórios do projeto back-end, é desenvolvido o serviço principal, que é as rotas, para o sistema foram desenvolvidas 3 rotas:

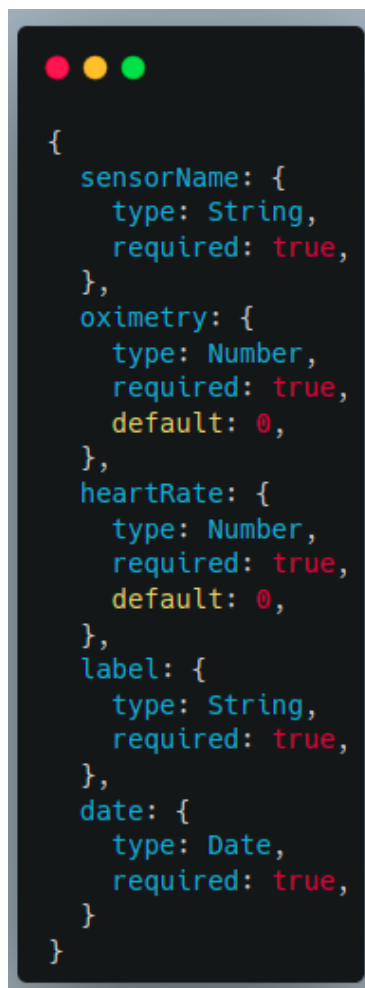
- POST “/register”: É uma rota do tipo POST com a função de realizar o cadastro de dados dos sensores no sistema.
- GET “/register/list-sensor”: É uma rota do tipo GET com a função de listar os sensores já cadastrados no sistema.
- GET “register/list-sensor/{nome-sensor}”: É uma rota do tipo GET com um parâmetro, ela possui a função de buscar o histórico de análises filtrando esse histórico a partir do nome do sensor.

3.11 Banco de dados MongoDB

Para a armazenagem do sistema, foi adotado o banco de dados MongoDB, pelo fato da sua performance e flexibilidade. Para utilização desse banco de dados a partir do back-end, foi necessário instalar a biblioteca “*mongoose*”, biblioteca responsável pela criação de esquemas e consultas deste banco a partir do back-end desenvolvido em

NodeJS. O banco possui dois esquemas semelhantes, mas com responsabilidades diferentes, o esquema register, com responsabilidade de armazenagem de histórico dos sensores, e o esquema sensor, com a responsabilidade de possuir somente sensores do sistema.

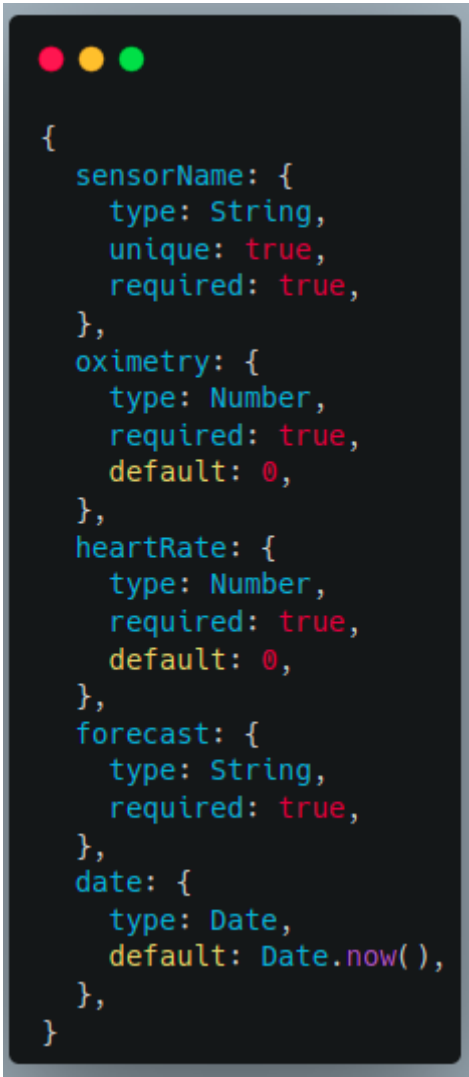
Para definir o que cada dado é no sistema, é necessário informar qual o seu tipo, além disso informar se o valor é único ou se é um campo obrigatório. O esquema sensor possui o campo *sensorName* como valor único, pois ele não permite cadastros de um mesmo sensor e possui o campo *date* com valor default para data atual, isso se dá pelo fato de ele registrar a data do momento em que foi realizado o cadastro do sensor. A Figura 17 exibe o esquema do banco para registros e a Figura 18 exibe o esquema do banco para sensores.



```
{
  sensorName: {
    type: String,
    required: true,
  },
  oximetry: {
    type: Number,
    required: true,
    default: 0,
  },
  heartRate: {
    type: Number,
    required: true,
    default: 0,
  },
  label: {
    type: String,
    required: true,
  },
  date: {
    type: Date,
    required: true,
  }
}
```

Figura 17 - Esquema para registros.

Fonte: Próprio autor.



```
{
  sensorName: {
    type: String,
    unique: true,
    required: true,
  },
  oximetry: {
    type: Number,
    required: true,
    default: 0,
  },
  heartRate: {
    type: Number,
    required: true,
    default: 0,
  },
  forecast: {
    type: String,
    required: true,
  },
  date: {
    type: Date,
    default: Date.now(),
  },
}
```

Figura 18 - Esquema para sensores.

Fonte: Próprio autor.

3.12 Projeto Front-end

Para o projeto da aplicação web que serve de interface do sistema para o usuário, foi planejado um fluxo de transição de telas que seguem uma ordem lógica, que vai desde a listagem do sensor até o histórico do mesmo. A Figura 19 mostra o fluxo de transição.

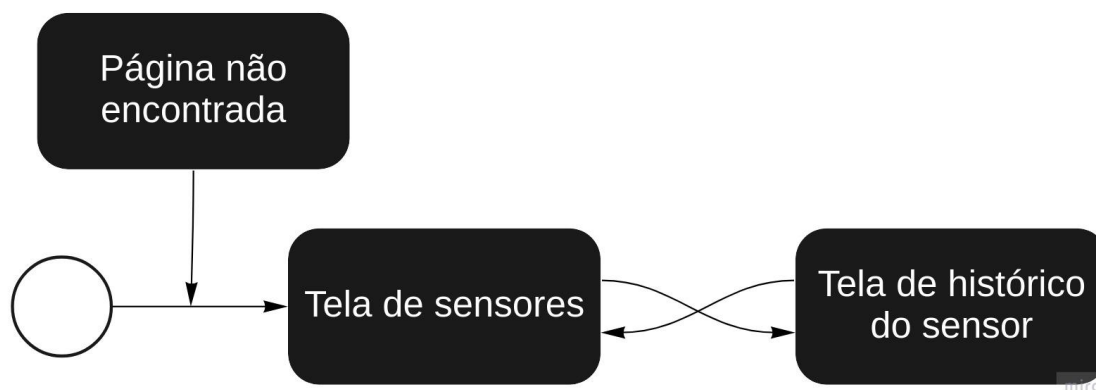


Figura 19 - Fluxo de transição de tela.

Fonte: Próprio autor.

Uma vez definido o fluxo, foi realizado o desenvolvimento das telas, com as interações com o back-end. Para que houvesse uma visualização em tempo real do sistema, foi aplicado a biblioteca “*socket.io*” responsável em fazer uma conexão em tempo real. A Figura 20 mostra a tela de sensores e a Figura 21 mostra a tela de histórico do sensor.

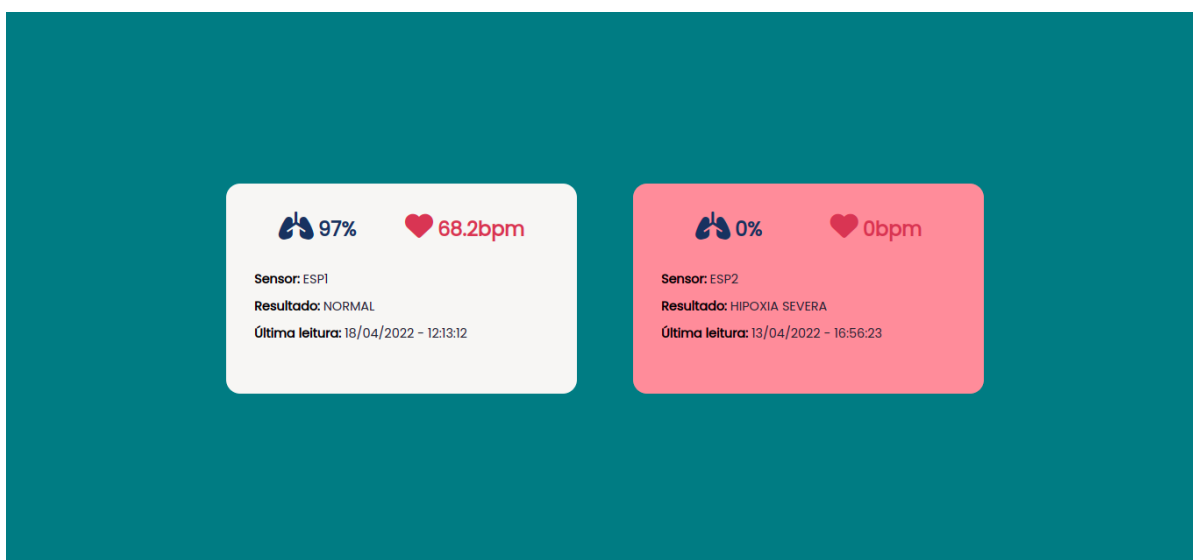


Figura 20 - Tela de sensores.

Fonte: Próprio autor.

Sensor	Frequência Cardíaca	Oximetria	Resultado	Data da Leitura
ESPI	68.2bpm	97%	NORMAL	18/04/2022 - 12:13:12
ESPI	73.33bpm	98%	NORMAL	18/04/2022 - 12:12:12
ESPI	53.03bpm	98%	NORMAL	18/04/2022 - 12:12:1
ESPI	34.62bpm	248%	NORMAL	18/04/2022 - 12:11:51
ESPI	80.24bpm	98%	NORMAL	13/04/2022 - 17:10:30
ESPI	73.01bpm	98%	NORMAL	13/04/2022 - 17:10:20
ESPI	69.05bpm	98%	NORMAL	13/04/2022 - 17:10:10
ESPI	68.8bpm	98%	NORMAL	13/04/2022 - 17:10:0
ESPI	79.38bpm	98%	NORMAL	13/04/2022 - 17:9:51
ESPI	18.76bpm	98%	NORMAL	13/04/2022 - 17:9:40
ESPI	72.9bpm	98%	NORMAL	13/04/2022 - 17:9:10
ESPI	75.16bpm	98%	NORMAL	13/04/2022 - 17:9:0
ESPI	79.49bpm	98%	NORMAL	13/04/2022 - 17:8:50
ESPI	65.26bpm	99%	NORMAL	13/04/2022 - 17:8:43
ESPI	65.01bpm	98%	NORMAL	13/04/2022 - 16:56:17
ESPI	69.6bpm	98%	NORMAL	13/04/2022 - 16:53:7
ESPI	68.19bpm	98%	NORMAL	13/04/2022 - 16:52:57
ESPI	71.71bpm	98%	NORMAL	13/04/2022 - 16:52:47
ESPI	73.7bpm	98%	NORMAL	13/04/2022 - 16:52:37

Figura 21 - Tela de histórico do sensor.

Fonte: Próprio autor.

A seguir é demonstrado na Figura 22 a estrutura de diretórios da aplicação web.

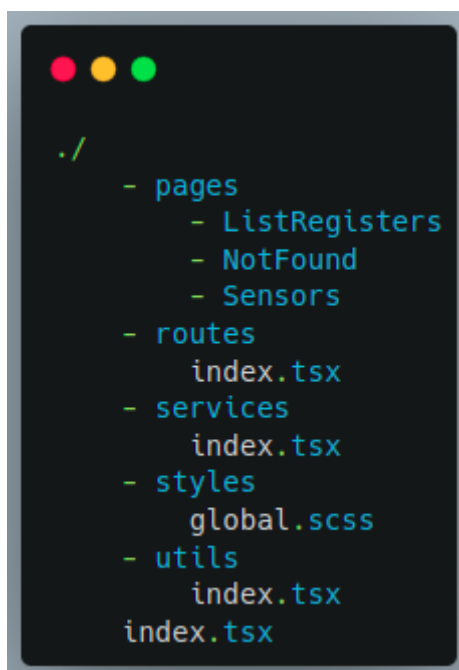


Figura 22 - Estrutura de diretórios do front-end.

Fonte: Próprio autor.

A seguir são explanadas as funções de cada diretório e dos arquivos responsáveis pelo funcionamento do projeto, são eles:

- `index.tsx`: assim como no back-end, esse é o arquivo principal no front-end, a única diferença está no formato do arquivo. O frontend está escrito em Typescript.
- `pages`: diretório onde se encontra todas as páginas do front-end.
 - `ListRegisters`: diretório onde contém os arquivos escritos para a renderização da tela de histórico do sensor.
 - `NotFound`: diretório onde contém os arquivos escritos para a renderização da tela não encontrada.
 - `Sensors`: diretório onde contém os arquivos escritos para a renderização da tela de sensores.
- `routes`: diretório onde contém o arquivo que administra as rotas do sistema, com propósito de renderizar uma tela de acordo com a requisição de um rota.
- `services`: diretório onde contém o DNS do back-end, para montar as requisições do sistema.
- `styles`: diretório onde contém todos os estilos de fonte, cores, formatos, entre outros, do sistema.
- `utils`: diretório onde contém algoritmos úteis para o sistema. No caso, o sistema conta com um método que realiza parser da data que vem em formato timestamp do banco, podendo assim reproduzi-la no formato DD/MM/YYYY - HH:MM:SS.

Uma vez definida a estrutura de diretórios do projeto front-end, é desenvolvido a nomenclatura das rotas, são elas:

- GET `"/`: é uma rota do tipo GET com a função de renderizar a tela de sensores.
- GET `"/list-registers"`: é uma rota do tipo GET com a função de renderizar a tela de histórico do sensor.
- GET `"*"`: é uma rota do tipo GET com função de renderizar a tela de página não encontrada. É `"*"`, pois significa que seja qualquer endereço

que o usuário colocar, que não esteja cadastrado no arquivo de routes, ele irá tratar como página não encontrada.

3.13 Bibliotecas utilizadas

Para a implementação do código de cada etapa do projeto, foram utilizadas bibliotecas específicas para estabelecer comunicação entre os dispositivos e para a implementação da interface de usuário.. As bibliotecas utilizadas foram:

- *Wire.h*: Utilizada para estabelecer a comunicação I2C entre os dispositivos;
- *MAX30100_PulseOximeter.h*: Utilizada para fazer interface do CI MAX30100, permitindo fazer alterações nos parâmetros do sensor e obter os dados de saturação do oxigênio e frequência cardíaca;
- *ESP8266WiFi.h*: Para fazer a conexão do ESP8266 com a internet;
- *PubSubClient.h*: Utilizado para o cliente MQTT na realização de *publisher* para transferência de dados;
- *paho.mqtt*: Utilizado para o cliente MQTT na realização de *subscribe* no python.
- *ArduinoJson.h*: Utilizado para montar a estrutura de dados para a transferência;
- *Mongoose*: Utilizada no *backend* para operar com as informações no banco de dados;
- *socket.io*: Utilizada para conexão web socket, para receber dados em tempo real.

4. RESULTADOS

Neste capítulo serão apresentados os resultados obtidos a partir dos testes realizados. É importante destacar que pela atual situação sanitária, os testes obtidos foram efetuados somente com o próprio autor e seus familiares, e para ratificar os resultados, foi utilizado um oxímetro de pulso comercial MedMax AS-301-L para a comparação dos resultados. Na figura 23 mostra o oxímetro de pulso comercial.



Figura 23 - Oxímetro MedMax AS-301-L.

Fonte: SENSOR MEDICAL.

4.1 Execução de testes nas rotas do back-end

Para verificar as respostas dos serviços a requisições de possíveis clientes, as rotas implementadas foram testadas utilizando o software Insomnia, que é um aplicativo desktop com objetivo de atuar como cliente para API's baseadas em HTTP.

A Figura 24 mostra o resultado da requisição de cadastro de dados do sensor, sua resposta é vazia, mas apresenta o status HTTP 201, que significa criação realizada com sucesso.

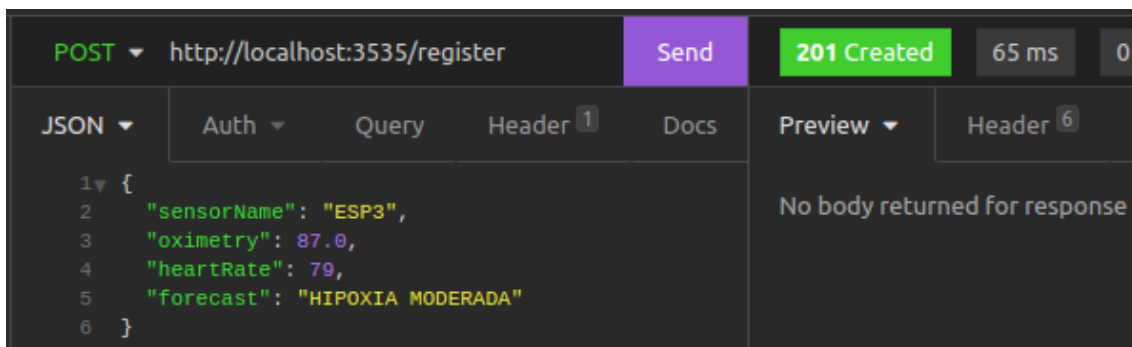


Figura 24 - Requisição para cadastro de sensor.

Fonte: Próprio autor.

A Figura 25 mostra o resultado da requisição de listagem de todos os sensores já cadastrados no sistema.

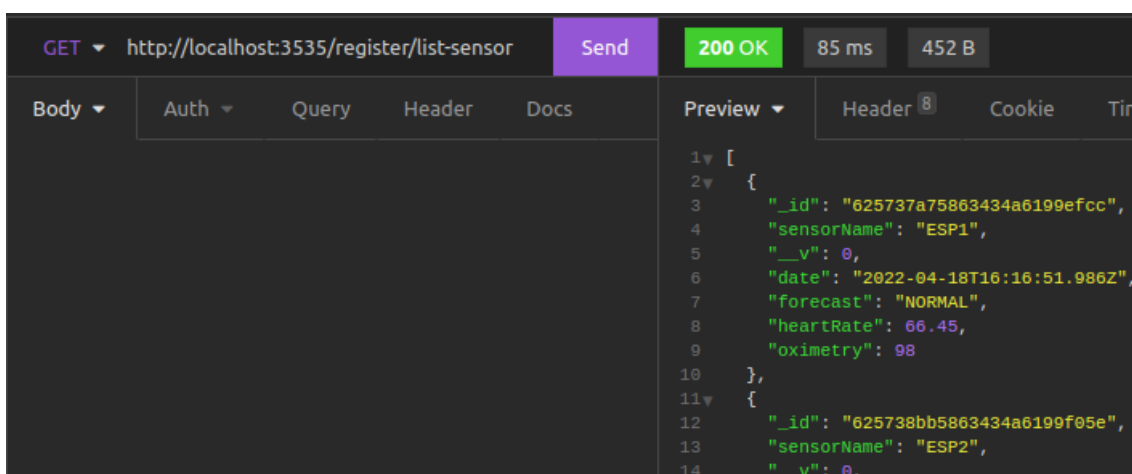


Figura 25 - Requisição para listagem de sensores.

Fonte: Próprio autor.

A Figura 26 mostra os resultados obtidos da requisição do histórico do sensor. Para especificar o tipo de sensor, é necessário que haja o envio do nome do sensor no parâmetro do endereço.

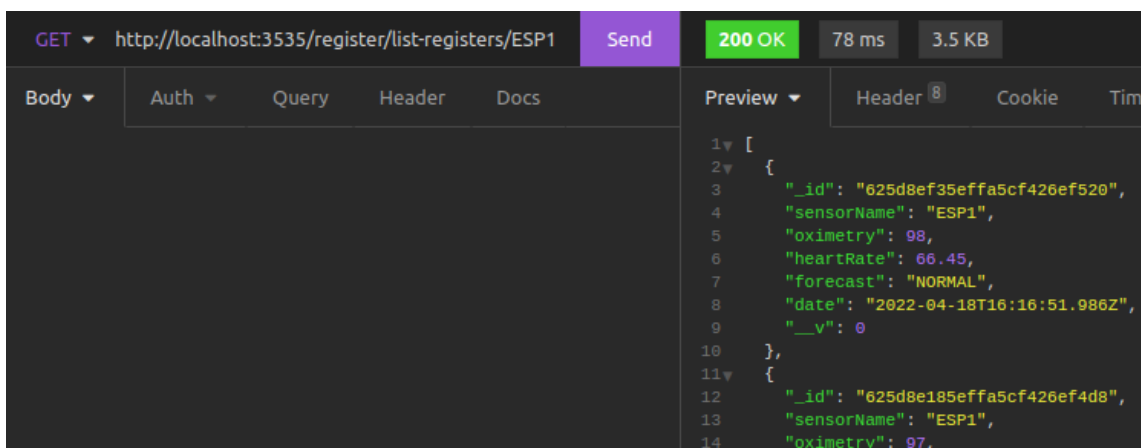


Figura 26 - Requisição para histórico do sensor.

Fonte: Próprio autor

4.2 Leituras

É possível afirmar que as leituras obtidas entre o oxímetro de pulso comercial comparado com as leituras do oxímetro do projeto, foram boas, pois os resultados foram bem próximos. Na Figura 27 mostra o resultado do oxímetro de pulso comercial realizado no próprio autor e a coleta do oxímetro comercial exibindo seu resultado na Figura 28, que exibe um resultado próximo dos resultados do oxímetro de pulso comercial, validando assim a precisão do sistema no momento da coleta.

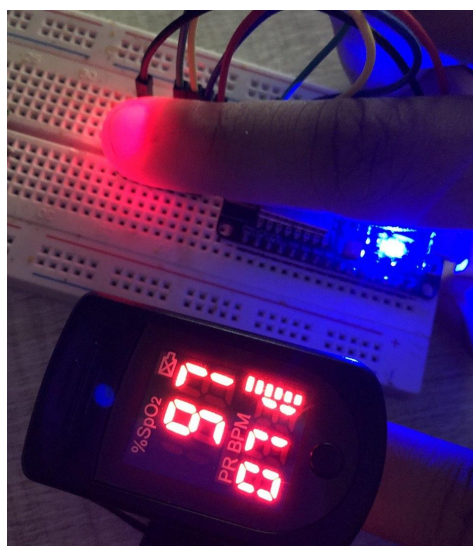


Figura 27 - Oxímetro de pulso comercial e coleta do oxímetro do sistema.

Fonte: Próprio autor.

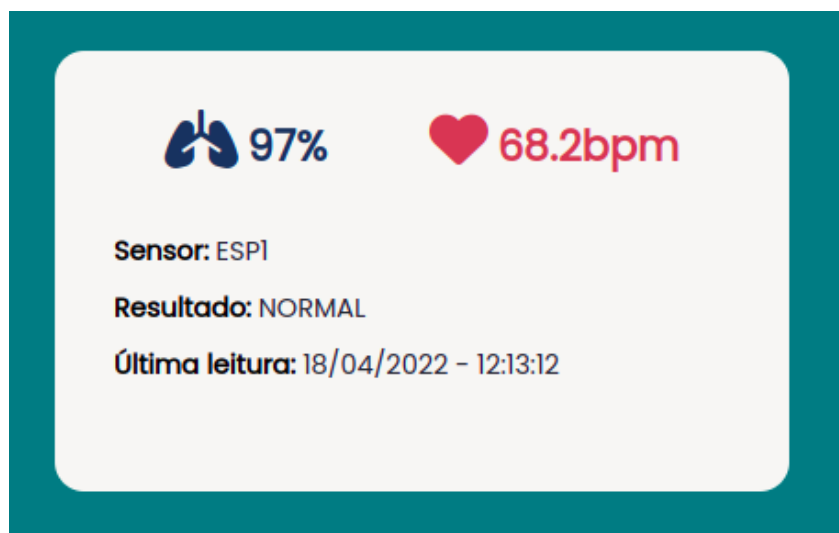


Figura 28 - Resultado oxímetro de pulso do projeto.

Fonte: Próprio autor.

4.3 Acurácia

Para ratificar a precisão do oxímetro de pulso do sistema com o oxímetro de pulso comercial, foi realizado o cálculo de acurácia, com amostra de 4 indivíduos, sendo um deles portador de marca-passo.

Indivíduo	Frequência Cardíaca (Comercial)	Oximetria (Comercial)	Frequência Cardíaca (Sistema)	Oximetria (Sistema)
Masculino adulto (portador marca-passo)	65 bpm	97%	66.5 bpm	98%
Masculino adulto	71 bpm	97%	68.2 bpm	97%
Feminino Adulto	67 bpm	98%	69 bpm	98%
Masculino > 10 anos	72 bpm	96%	74.5 bpm	97%

Tabela 1 - Resultado coletas

Fonte: Próprio autor

Para mensurar a acurácia, será realizado a média de cada dado coletado, frequência cardíaca e oximetria, e assim aplicando na Equação 4, que é a equação do cálculo de acurácia.

$$Acurácia = (média\ dados\ concretos / média\ dados\ a\ validar) * 100 \quad (4)$$

Aplicando os valores das frequências cardíacas nos campos da Equação 4, temos a Equação 5, apresentando os resultados da precisão da frequência cardíaca do sistema.

$$Acurácia = (68,75 / 69,55) * 100 \simeq 98,8\% \quad (5)$$

E por fim, aplicando os valores das oximetrias nos campos da Equação 4, temos a Equação 6, apresentando os resultados da precisão da oximetria do sistema.

$$Acurácia = (97 / 97,5) * 100 \simeq 99,48\% \quad (6)$$

4.4 Coletas simultâneas

Foram realizadas as coletas simultâneas no projeto, e obteve-se resultados simultâneos em tempo real, conforme explicita a Figura 29, que mostra as duas coletas sendo realizadas pelo oxímetro de pulso do sistema, e na Figura 30, os seus resultados no sistema web.

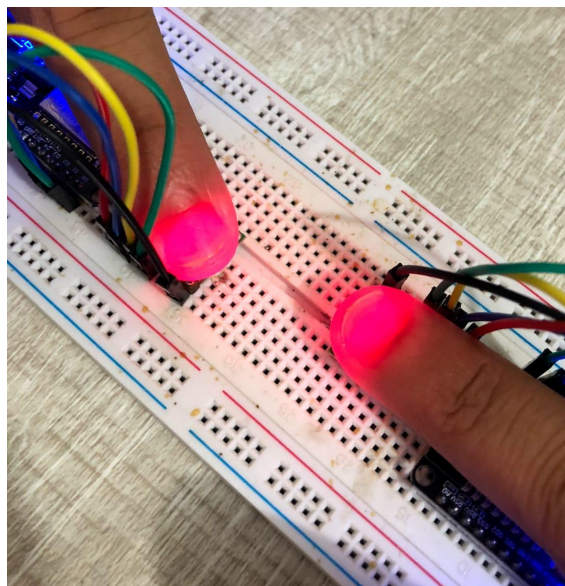


Figura 29 - Coletas simultâneas

Fonte: Próprio autor



Figura 30 - Resultado coletas simultâneas

Fonte: Próprio autor

4.5 Interface Sistema Web

Na interface web, foi levado em consideração os seguintes dados: nome do dispositivo que está coletando, frequência cardíaca, oximetria, resultado da rotulagem de oximetria e a data da coleta.

O resultado da rotulagem é expresso tanto em escrita na informação do card como também com as mudanças de cores, de tal modo que tome a atenção do profissional da saúde para o resultado da coleta de um determinado paciente.

A Figura 31, mostra o comportamento do card com o resultado obtido com a hipoxia leve.

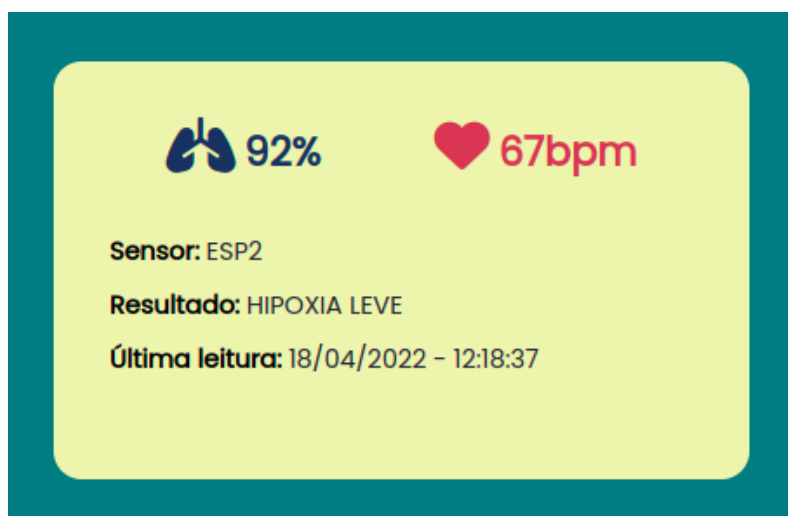


Figura 31 - Cor leve dos resultados de rotulagem.

Fonte: Próprio autor.

A Figura 32, mostra o comportamento do card com o resultado obtido com a hipoxia moderada.

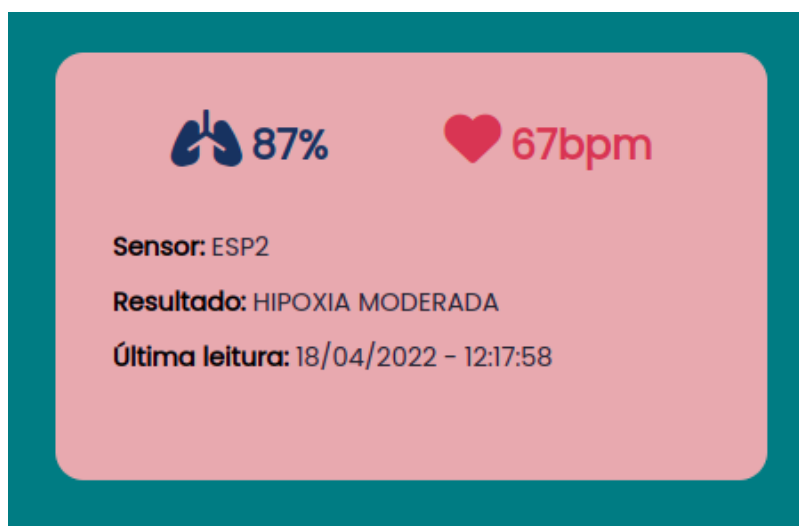


Figura 32 - Cor moderada dos resultados de rotulagem.

Fonte: Próprio autor.

A Figura 33, mostra o comportamento do card com o resultado obtido com a hipoxia severa.

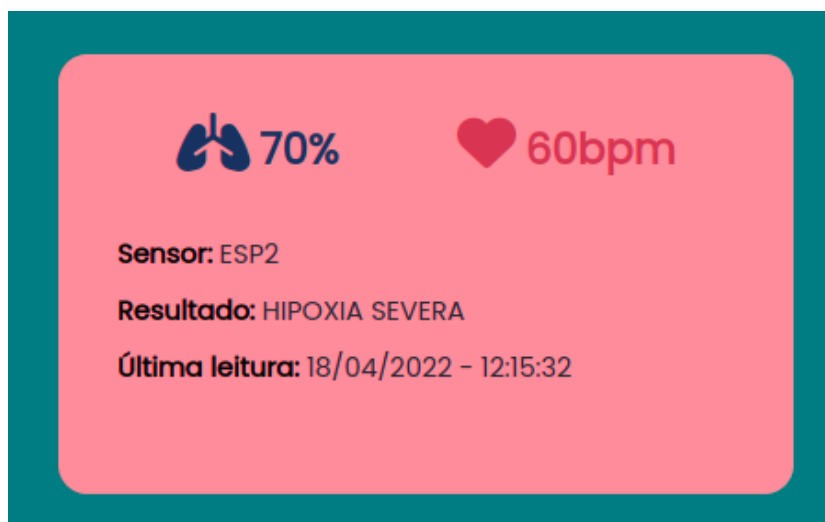


Figura 33 - Cor severa dos resultados de rotulagem.

Fonte: Próprio autor.

No projeto, os testes foram realizados com apenas dois sensores, mas não se limita a isso, pois da forma que foi criada a arquitetura, é possível ter bem mais de dois sensores, podendo assim ser aplicado a um cenário de ambiente hospitalar. Na Figura 34 demonstra três cards de sensores, os dois primeiros brancos são o rastreamento em tempo real, o terceiro não possui o rastreamento de sensores, pois este foi imputado manualmente no banco de dados, com o propósito de demonstrar o resultado exibido ao usuário quando há situações críticas.

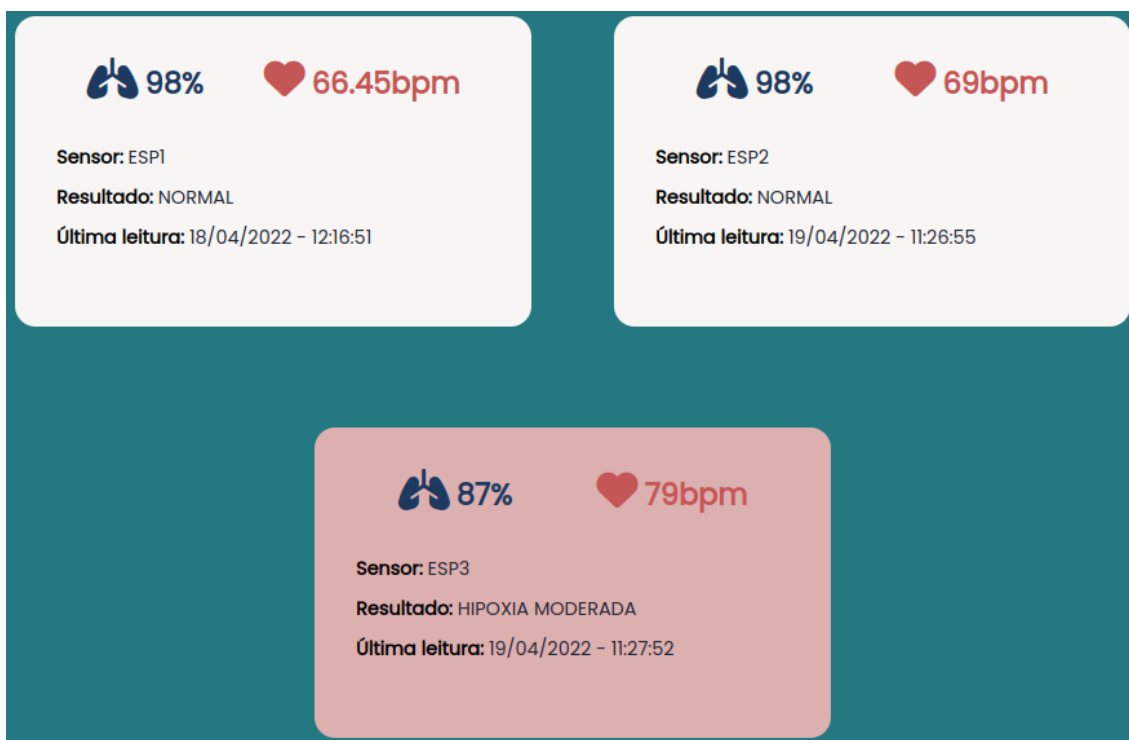


Figura 34 - Listagem de sensores.

Fonte: Próprio autor.

A Figura 35 demonstra todo o histórico do primeiro sensor, neste é possível ver a variação da frequência cardíaca, a oximetria, seu resultado e a data da leitura de cada teste. Ajudando o profissional da saúde a fazer análises do progresso do paciente.

Sensor	Frequência Cardíaca	Oximetria	Resultado	Data da Leitura
ESPI	66.45bpm	98%	NORMAL	18/04/2022 - 12:16:51
ESPI	68.2bpm	97%	NORMAL	18/04/2022 - 12:13:12
ESPI	73.33bpm	98%	NORMAL	18/04/2022 - 12:12:12
ESPI	53.03bpm	98%	NORMAL	18/04/2022 - 12:12:1

Figura 35 - Histórico de leituras.

Fonte: Próprio autor.

5. CONCLUSÕES

5.1 Considerações Finais

Nesta monografia, desenvolveu-se um sistema completo, composto por *hardware* e *software*, a partir de uma demanda focada no monitoramento de oximetria de diversos pacientes simultâneos a partir de qualquer localização, possibilitando análise e tomadas de decisões para os profissionais da saúde.

Os componentes físicos MAX30100, ESP8266 NodeMCU e *Raspberry Pi 2* e os frameworks NodeJS e ReactJS escolhidos para o desenvolvimento *web*, mostraram-se ideais dentro do que se propõem, não apresentando curvas acentuadas de aprendizado, ou dificuldade na implementação.

A considerar a aplicação desenvolvida e sua potencialidade, é possível concluir que o sistema apresentado pode ser aplicado tanto no ambiente hospitalar, como também no ambiente de atendimento remoto, possibilitando abrir portas para mais implementações na área da saúde, e assim prevenindo situações de eventos não previstos com antecedência.

5.2 Trabalho Futuro

Este projeto propôs uma coleta de dados em tempo real por meio do oxímetro de pulso, e há outras formas de aplicação para complementar este trabalho e deixar, são elas:

- Implementar inteligência para chamadas de emergência de acordo com os dados graves obtidos, para usuários que fazem monitoramento domiciliar.
- Implementar coleta de dados vitais com análises de pressão arterial, temperatura corporal e respiração, e assim exibir em tempo real para o profissional da saúde.
- Implementar técnicas de inteligência artificial, como aprendizado de máquina, prever evolução clínica dos pacientes a partir de dados coletados pelos sensores, por meio do monitoramento remoto ou triagem clínica.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] MURTA, G. F; et al. **Saberes e Práticas: guia para ensino e aprendizado de enfermagem**. 5. ed. São Caetano do Sul: Difusão, 2009.
- [2] MEDEIROS, Ana Lúcia et al. **Oximetria de pulso em triagem de cardiopatias congênitas: conhecimento e atuação do enfermeiro**. Cogitare Enfermagem, v. 20, n. 3, 2015.
- [3] LUKS, Andrew M; SWENSON, Erik R. **Pulse Oximetry for monitoring patients with covid-19 at home: Potential pitfalls and practical guidance**. Annals of the American Thoracic Society, n. ja, 2020.
- [4] PIRES, Luiza Nassif; CARVALHO, Laura; XAVIER, Laura de Lima. **Covid-19 e desigualdade: a distribuição dos fatores de risco no Brasil**. Experiment Findings, 2020
- [5] PALHA, Álvaro. **Colapso do sistema de saúde de Manaus**. Disponível em: <<https://www.msf.org.br/diarios-de-bordo/colapso-do-sistema-de-saude-de-manaus/>>, 2021. Acesso em: 15 de Abril de 2022.
- [6] PEREDNIA, Douglas A; ALLEN, Ace. **Telemedicine technology and clinical applications**. Jama, American Medical Association, v. 273, n. 6, p. 483 - 488, 1995.
- [7] HALL, John Edward; GUYTON, Arthur C. Tratado de fisiologia médica. In: **Tratado de fisiologia médica**. [S.I.: s.n.], 2011. p. 1151 - 1151.
- [8] MEHRABI, Mohsen. **Design of a new reflectance pulse oximeter by obtaining the optimal source-detector space**. Optic, vol. 168, p. 34-45, September, 2018.
- [9] GARZÓN, Maurício F. et al. **Fisioterapia respiratoria en fibrosis pulmonar por COVID-19**. Revista Espacios. vol. 43, n. 3, Março, 2022.

[10] CALIL, Saide Jorge. et al. **Equipamentos médico-hospitalares e o gerenciamento da manutenção**. Ministério da Saúde. Brasília, p. 0, 2002.

[11] ELGENDI, Mohamed. **On the analysis of fingertip photoplethysmogram signals**. Current cardiology review, Bentham science PUBLISHERs, v. 8, n. 1, p. 14-25, 2012.

[12] BARCH DAILY. **Novo aplicativo do Morpholio Project mede a reação humana ao ambiente construído**. 2014. Disponível em: <<https://www.archdaily.com.br/br/602151/novo-aplicativo-do-morpholio-project-mede-a-reacao-humana-ao-ambiente-construido/53613156c07a80d43c000137>>, Acesso em: 15 de Abril de 2022.

[13] SANTOS, Bruno P. et al. **Internet das coisas: da teoria à prática**. Minicursos SBRC-Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, v. 31, 2016.

[14] FERNANDES, Jorge H. C. **Qual a prática do desenvolvimento de software**. Ciência e Cultura, vol. 55, no. 2, São Paulo, Abril/Junho 2003.

[15] OPUS SOFTWARE. **Node.js - O que é, como funciona e quais as vantagens**. 2018. Disponível em: <<https://www.opus-software.com.br/node-js/>>. Acesso em: 15 de Abril de 2022.

[16] KENZIE. **React: O que é, como funciona e porque usar e como aprender**. 2020. Disponível em: <<https://kenzie.com.br/blog/react/>>. Acesso em: 15 de Abril de 2022.

[17] TREINAWEB. **O que é MongoDB**. 2021. Disponível em: <<https://www.treinaweb.com.br/blog/o-que-e-mongodb#:~:text=%C3%89%20um%20banco%20de%20dados,desempenho%20e%20facilidade%20para%20consultas.>>>. Acesso em: 15 de Abril de 2022.

[18] MAX30100 Datasheet; **Pulse oximeter and Heart-Rate Sensor IC for Wearable Health**, 2014.

[19] LABORATÓRIO DE GARAGEM. 2019. Disponível em: <<https://labdegaragem.com/forum/topics/ox-metro-max30100?commentId=6223006%3AComment%3A721563>>. Acesso em: 15 de Abril de 2022.

[20] ESP8266 Datasheet. **ESP8266EX Datasheet**. Espressif Systems, version 6.6, 2020.

[21] PISCALED. **Módulo WiFi ESP8266 NodeMCU**. Disponível em: <<https://www.piscaled.com.br/modulo-wifi-esp8266-nodemcu-esp12f-cp2102-amica>>. Acesso em: 15 de Abril de 2022.

[22] RASPBERRY PI. 2015. Disponível em: <<https://www.raspberrypi.com/products/>>. Acesso em: 15 de Abril de 2022.

[23] EMBARCADOS. **Raspberry Pi 2 Modelo B**. 2015. Disponível em: <<https://www.embarcados.com.br/lancamento-raspberry-pi-2-model-b/>>. Acesso em: 15 de Abril de 2022.

[24] THE IOT PROJECTS. **ESP32 based MAX30100 pulse oximeter webserver**. 2021. Disponível em: <<https://theiotprojects.com/esp32-based-max30100-pulse-oximeter-webserver/>>. Acesso em: 15 de Abril de 2022.

[25] SENSOR MEDICAL. **Oxímetro de dedo**. Disponível em: <<https://www.sensormedical.com.br/departamento/80763/0102/oximetro-de-dedo>>. Acesso em: 15 de Abril de 2022.