

Dispositivo para monitoramento e acompanhamento de sinais vitais em ambientes hospitalares: desenvolvimento de um protótipo

Mateus Felipe Pscheidt.¹

Fabio Fernando Kobs, Dr.²

RESUMO: A *internet* das coisas é uma tecnologia que pode ser implementada em diversos setores, inclusive na saúde. Em ambientes hospitalares há necessidade de aferição de sinais vitais, bem como a temperatura corporal, batimentos cardíacos e até mesmo a oxigenação do sangue em intervalos de tempo na maioria dos pacientes. Tendo em vista a aferição contínua, tem-se como problema a ausência de um dispositivo para realizar as aferições a cada intervalo de tempo (previamente definido), e com o envio dos dados aferidos para um servidor. Ainda, considera-se como problema a ausência de controles de pacientes específicos, de grupos, dentre os sinais vitais. Nesse sentido, o objetivo geral é desenvolver um protótipo de *hardware* que será “conectado” ao paciente para realizar as aferições, principalmente em um grupo de pessoas, no qual não há necessidade de intervenção de um profissional para tais aferições. Diante da evolução da comunicação entre redes sem fio (*wifi*), e por meio de outras tecnologias, bem como os sensores, buscou-se resolver o problema com a utilização de microcontrolador, sensores, com o uso de sistema embarcado e sistema *Web*. A partir da *internet* das coisas e com o desenvolvimento do protótipo, ou seja, do dispositivo para monitoramento de pacientes em hospitais, propôs-se uma maneira de aferição e acompanhamento dos sinais vitais do corpo humano por meio das tecnologias envolvidas, resultando em uma ferramenta de acompanhamento do paciente com interface WEB.

Palavras-chave: *Internet* das coisas; *Wearable*; ESP8266, Sistema embarcado.

1. INTRODUÇÃO

A *Internet* das Coisas (IoT³) é uma tecnologia que pode ser implementada em diversos setores, inclusive na saúde, sendo possível conectar diversos dispositivos em uma rede sem fio, assim podendo enviar/receber dados para um determinado ponto, quer seja um roteador, servidor, ou até mesmo um outro dispositivo.

Em ambientes hospitalares há grande necessidade de aferição dos sinais vitais, bem como a temperatura corporal, batimentos cardíacos e até mesmo a oxigenação do

¹ Artigo apresentado no Curso de Sistemas de Informação como requisito parcial à obtenção do título de Bacharel em Sistemas de Informação da Universidade do Estado de Santa Catarina no ano de 2023.

² Professor orientador.

³ IoT (Internet of Things), a *internet* das coisas tem objetivo de conectar e trocar dados com outros dispositivos e sistemas pela *Internet* (ORACLE, 2023)

sangue a cada intervalo de tempo em pacientes. Nesse sentido, o objetivo está na agilidade das aferições, principalmente em um grupo de pessoas, no qual demandaria muito tempo, além de não haver necessidade de intervenção de um profissional para estas aferições, pois através deste dispositivo, as aferições serão de modo automáticos e contínuos.

O uso da tecnologia na saúde é inevitável, com o avanço tecnológico atual, é possível auxiliar todos os envolvidos, principalmente quando há necessidade de aferir os sinais vitais, como a temperatura corporal, batimentos cardíacos entre outros. Nestes casos, há problemas com os toques físicos (maior possibilidade de transmitir doenças) e também o tempo gasto pelos profissionais, podendo, assim, deixar este importante tempo dos profissionais com atividades que não é possível automatizar.

Nesse contexto, as tecnologias podem auxiliar, assim como no desenvolvimento deste *hardware*. Diante disso, acredita-se que a utilização do dispositivo (principalmente em massa) poderá diminuir o tempo gasto pelos profissionais e pela sua utilidade, principalmente pelo monitoramento de diversos pacientes ao mesmo tempo, alertando somente para casos adversos configurados previamente.

Nesse sentido, tem-se como objetivo principal desenvolver um dispositivo de monitoramento de saúde no ambiente hospitalar baseado em IoT que registra a temperatura corporal e outros parâmetros. De modo prático, o dispositivo se conecta a uma rede *wifi* previamente configurada, envia estas informações ao servidor local, por meio de HTTP, isso ocorre a cada certo intervalo de tempo pré-definido, por exemplo, a cada 1 segundo, podendo ser visualizadas no sistema local quando necessário.

A motivação para a decisão deste desenvolvimento, encontra-se principalmente pela contribuição tecnológica aos setores da saúde de modo a melhorar e acelerar os processos básicos e contínuos, bem como a melhor avaliação de resultados obtidos durante o tratamento de pacientes. Além, também, do aprofundamento das tecnologias existentes, para a implementação deste protótipo.

Na prática, o dispositivo deve ser utilizado de forma que os sensores fiquem no punho, pois este é o local onde os sensores devem realizar as leituras das informações, e dos sinais vitais. Por se tratar de componentes pequenos e leves, os sensores e a placa de controle ficam juntas, tornando assim o dispositivo um *wearable*.

O artigo encontra-se estruturado da seguinte forma: o capítulo 2 apresenta a referência teórico, trabalhos relacionados, comunicação, materiais e métodos utilizados. Já no capítulo 3 é apresentada o protótipo do *hardware*. O capítulo 4 apresenta o protótipo do *software* e, finalmente, o capítulo 5 traz as conclusões.

2. REFERENCIAL TEÓRICO E ESTADO DA ARTE

Este capítulo apresenta: trabalhos relacionados, HTTP, materiais e métodos utilizados.

2.1 Trabalhos Relacionados

Há outros estudos que buscaram possibilidades em IoT para hospitais, principalmente sobre dispositivos com diversas funções, sendo um *wearable*. Um Sistema e implementação com MQTT⁴ foi proposto a partir de um protocolo em saúde baseado em sistemas IoT (TAREQ; KHALEE, 2021). O estudo utilizou apenas um sensor de temperatura corporal, e observa-se que a tecnologia (MQTT) utilizada necessita de internet para funcionar (TAREQ; KHALEE, 2021).

Outro caso que pode ser levado em conta, é um dispositivo que mede a temperatura corporal baseado em ESP8266⁵, proposto por Boulezhar, Abouricha e Kasmi (2021). Neste caso também se implementou apenas a aferição de temperatura, porém sem o MQTT e utilizando o ThingSpeak, que é um software para realizar a comunicação do mesmo, e que da mesma forma, é necessário ter acesso a internet. (BOULEZHAR; ABOURICHA; KASMI, 2021).

Uma outra alternativa a ser ponderada é a questão da segurança, conforme destacado por Abbas Abd Ali, Adnan Hussein Ali e Ali J Al-Askery (2020). Contexto no qual, optou-se por utilizar servidores da Amazon, onde os mesmos estão disponíveis na internet, ou seja, não são locais, o seu uso é devido principalmente ao elevado nível de confiabilidade, e também, de muitos acessos simultâneos. (ALI; ALI; AI-ASKERY, 2020).

Assim como documentado, (Arduino, 2018), o Arduino é uma plataforma de código aberto que oferece uma série de placas de circuito integrado e um ambiente de programação integrado (IDE), permitindo a criação de projetos eletrônicos interativos. Essa plataforma é capaz de receber *inputs* de diversos componentes, incluindo sensores, e realizar *outputs* para realizar funções específicas. Em resumo, o Arduino é uma ferramenta valiosa para os desenvolvedores de projetos eletrônicos, possibilitando a criação de soluções personalizadas para diversas aplicações. (ARDUINO, 2018).

O NodeMCU contém um *firmware open source* baseado no próprio ESP8266, conforme citado na documentação do NodeMCU (2023), com esse *firmware* de fábrica é possível programar na linguagem Lua, ou também é possível alterar para outras linguagens de programação, bem como o C++. (NodeMCU Documentation, 2023).

⁴ MQTT: MQTT (Message Queuing Telemetry Transport) é um protocolo de mensagens padrão OASIS para IoT. (MQTT, 2023).

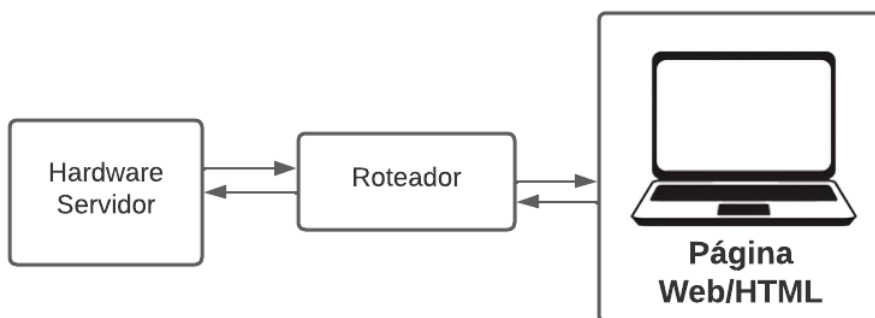
⁵ ESP8266: É um chip integrado e projetado para fornecer conectividade (ESP8266, 2023).

O ESP8266 permite que se execute os programas carregados em sua memória interna, bem como descrito por Oliveira (2017), desta maneira o ESP8266 funciona de forma embarcada, sem depender de outros dispositivos externos, sendo necessário apenas os sensores escolhidos. (OLIVEIRA, 2017).

2.2 HTTP

A comunicação entre o dispositivo de aferição e o roteador/servidor ocorrerá por meio do protocolo HTTP⁶, que é um protocolo de transferência de dados amplamente utilizado na *Internet*. O dispositivo de aferição enviará os dados para o servidor em uma solicitação HTTP. O servidor processará os dados recebidos e tratará de forma necessária para o usuário ler as informações, conforme demonstrado na Figura 1.

Figura 1 – Diagrama HTTP



Fonte: Elaborado pelos autores (2023).

A comunicação se dá através da rede *wifi* local, na qual é utilizada para enviar os dados para um servidor (local), podendo futuramente configurar um servidor na nuvem, porém, caso seja utilizado um servidor em nuvem, há necessidade de tomar um cuidado maior com a segurança na rede e os respectivos dados, pois a rede em questão estará mais suscetível a ataques e vazamentos de dados.

“Em se tratando de segurança da informação é sempre importante lembrar que esta abrange um conjunto de medidas que envolve, entre outros fatores, procedimentos técnicos.”, ou seja, o mínimo necessário a se fazer, é utilizar os métodos básicos de segurança na rede sem fio *wifi*. (RUFINO, 2005).

Por se tratar de um servidor local e sem acesso à *internet* (apenas rede intranet), pode ser considerada mais fácil de controlar, pois a exposição de ameaças externas é

⁶ HTTP: HyperText Transfer Protocol é um protocolo de camada de aplicação (HTTP, 2023)

muito limitada, além das redes *wifi* utilizadas serem criptografadas, para evitar assim diversos tipos de ameaças.

2.3 ESP8266

O ESP8266 é uma placa de desenvolvimento *open-source*⁷, modelo escolhido para este projeto, pois além de se conseguir encontrar facilmente no mercado, este modelo é de baixo custo, o que auxilia no desenvolvimento. O ESP8266 possibilita adicionar diversos outros tipos de *hardwares* (como os sensores), sendo que eles podem funcionar simultaneamente. A versão escolhida, possui *chip* de *wifi* integrado, que facilita, pois em uma única placa é possível se conectar à *internet*, receber os sinais dos sensores externos por meio de entradas e saídas analógicas/digitais, além de processar as informações recebidas e enviá-las ao servidor configurado. (EMBARCADOS, 2023).

De modo geral, uma placa de ESP8266 é muito semelhante à de um Arduino, além de ser programada com a mesmo ambiente de desenvolvimento. “A placa Arduino é muito similar à de um computador de pequeno porte, sendo composto por um microcontrolador, memória RAM, armazenamento secundário (memória *flash*) e *clock*, entre outras funcionalidades.” (OLIVEIRA, 2015). Este tipo de dispositivo, é amplamente utilizada para protótipos e dispositivos de IoT. (OLIVEIRA, 2015).

O ESP8266 é um *hardware* microcontrolador de baixo custo e baixo consumo de energia, desenvolvido para se conectar em redes *wifi*. Podendo executar diversas tarefas em tempo real, assim como leitura de dados dos sensores, envio destes dados aos servidores previamente configurados. O ESP8266 pode ser programado em várias linguagens de programação, bem como o C++ e LUA.

2.4 Sensores

Para realizar o monitoramento de frequência cardíaca e oxigenação do sangue, é necessário um sensor externo, para este fim, o protótipo conta com o sensor MAX30100 que possui um fotoacoplador otimizado, e quando conectado com um processador analógico como o ESP8266, é possível realizar as leituras de batimentos cardíacos e oxigenação do sangue do paciente. O protótipo conta também com um sensor de temperatura corporal do tipo NTC (10K), no qual faz a medição da temperatura, o ESP8266 recebe as informações de valores do sensor e calcula a temperatura.

A hemoglobina oxigenada (HbO₂) no sangue arterial tem a característica de absorver a luz do tipo infravermelho, bem como descrito em *Last Minute Engineers* (2023), quanto mais vermelho o sangue (maior a quantidade de hemoglobina presente no sangue), maior a quantidade de luz infravermelha é absorvida. Conforme o sangue

⁷ Open-source: é um software que contém código fonte no qual qualquer usuário pode inspecionar, modificar ou aprimorar. (OPENSOURCE, 2023).

vai sendo bombeado pelo dedo (a cada batimento cardíaco), desta forma a quantidade de luz refletida altera, desta maneira o sensor (fotodetector) consegue criar uma variável (com onda). Também é possível afirmar que a oximetria é calculada verificando a quantidade de luz vermelho e infra vermelho é absorvida, no qual varia dependendo da quantidade de oxigênio existente no sangue. (LAST MINUTE ENGINEERS, 2023).

A partir do estado da arte e da identificação das características, sobretudo o microcontrolador ESP8266 e sensores MAX30100 e NTC 10k. Na sequência encontram-se descritas as principais etapas para o desenvolvimento do protótipo do hardware e do software.

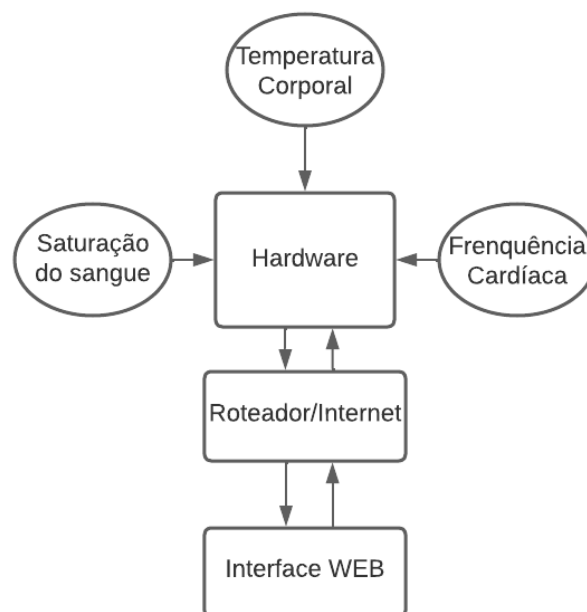
Nos capítulos de protótipos de *hardware* e de *software*, serão apresentadas as características e a visão geral do projeto (com mais detalhes), bem como o desenvolvimento, desde a escolha do microcontrolador e dos respectivos sensores, bem como o desenvolvimento do *software* e sua lógica.

2.5 Materiais e Métodos Utilizados

No diagrama de *hardware* (Figura 2), é mostrado de forma geral, as partes do dispositivo, assim como suas conexões, que está dividido em três partes, sendo:

- O *hardware* com seus respectivos sensores de temperatura corporal e de frequência cardíaca/oxigenação sanguínea.
- O servidor local, que se dá pelo ponto de acesso ou o roteador.
- Interface WEB, na qual o utilizador irá ver as informações pertinentes.

Figura 2 – Diagrama de Hardware.



Fonte: Elaborado pelos Autores (2023).

Os três tipos de sensores (Temperatura, saturação sanguínea e também de frequência cardíaca), são interligados no microcontrolador ESP8266 por meio de entradas e saídas analógicas, no qual irá processar as informações e enviá-las para um servidor local por meio de HTTP e executadas por um HTML⁸. No computador cliente conectado na mesma rede local, é possível estabelecer uma comunicação onde pode-se visualizar as informações em tempo real.

Figura 3 – Dispositivo de *hardware* utilizado.



Fonte: Elaborado pelos Autores (2023).

A Figura 3 demonstra o ESP8266, sensor de Batimentos cardíacos/oxigenação e também o sensor de temperatura (NTC), respectivamente.

Embora o ESP8266 não seja um Arduino, a IDE, bibliotecas, comunicações e outros, são iguais à de um Arduino, de modo geral o que há de diferença entre o Arduino e de um ESP8266, é a conectividade e o tamanho. Dessa forma, é possível utilizar as funcionalidades do ESP8266, como a conectividade *wifi*, em projetos de um Arduino. No caso deste projeto, o ESP8266 é fundamental, pois é menor que um Arduino, mais acessível, e ainda conta com *wifi*.

A parte do *hardware*, ainda pode contar com uma bateria, para a sua utilização, e assim não depender de quaisquer tipos de cabos, ou alimentação externa, sendo que, certamente a bateria deve ser recarregada, ou substituída a cada intervalo de tempo.

3. PROTÓTIPO DO HARDWARE

Inicialmente identificou-se os melhores componentes eletrônicos para integrar o *hardware* do dispositivo, levando em consideração o custo benefício. Em um primeiro momento optou-se pelo uso de um ESP8266, principalmente pela sua versatilidade, baixo custo, rede *wifi* integrada e baixo consumo de energia (intenção de utilizar com bateria no futuro). Em relação aos sensores, foi adicionado à placa principal (ESP8266)

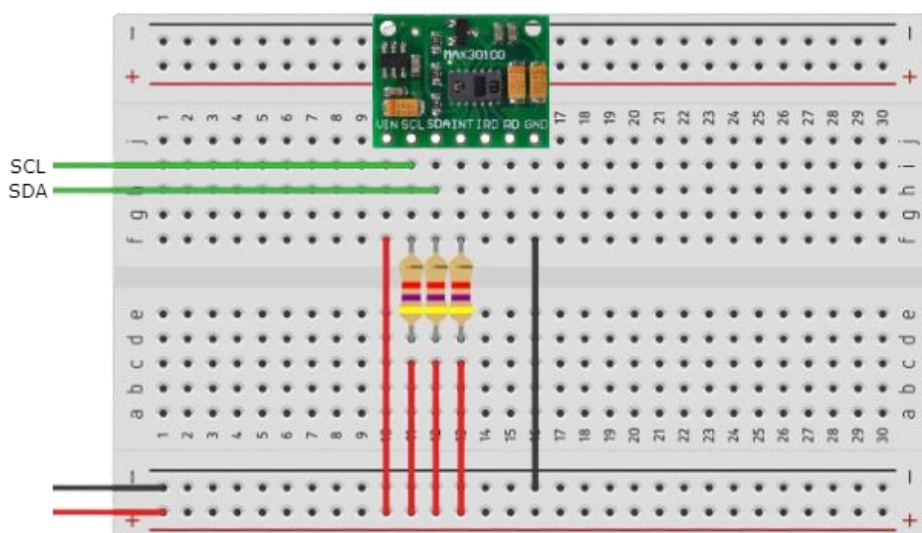
⁸ HTML: Hypertext Markup Language (HTML, 2023)

o MAX30100, no qual realiza a leitura dos batimentos cardíacos e oxigenação do sangue do paciente, e também um sensor de temperatura NTC (Figura 7).

Durante a montagem do protótipo (*hardware*), e no início dos testes, o sensor MAX30100 não estava funcionando corretamente, depois de uma série de testes e pesquisas, foi identificado que o problema não estava na montagem, e sim no próprio sensor. Há um problema no esquema do sensor MAX30100 na questão da alimentação de energia. O *chip* do MAX30100 requer uma alimentação de 1,8v, este módulo utiliza dois reguladores para atingir esta tensão, porém esta alimentação não chega corretamente no chip, podendo gerar problemas na comunicação com o Arduino (que trabalha com níveis lógicos).

Para resolver este problema, foi removido os três resistores do sensor (os quais controlavam a tensão) e adicionado outros três resistores de 4.7k Ω , conforme figura 4, com isso a alimentação necessária chega no *chip*, conseqüentemente a comunicação funciona normalmente com o ESP8266 (através dos pinos SCL e SDA).

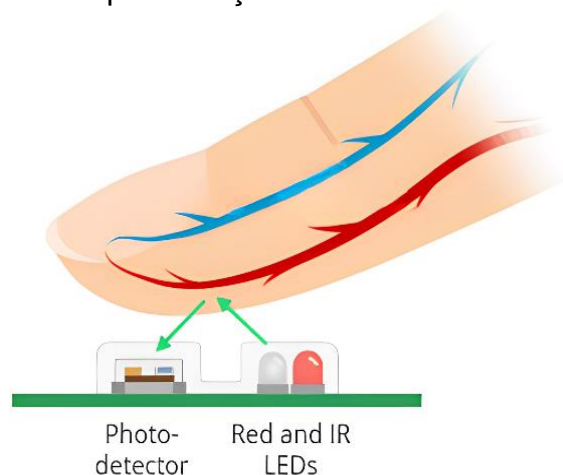
Figura 4 – Diagrama (com os resistores adicionados)



Fonte: Acervo dos próprios autores (2023).

O sensor MAX30100 utiliza dois LEDs – um deles é vermelho e o outro é infravermelho que pode emitir uma luz através do tecido do dedo do paciente, onde o sensor fotodiodo mede a quantidade de luz (vermelha e infravermelho) é absorvida pelo sangue, desta forma é possível determinar o nível de oxigênio do sangue e o batimento cardíaco (Figura 5).

Figura 5 – Representação funcionamento MAX30100.

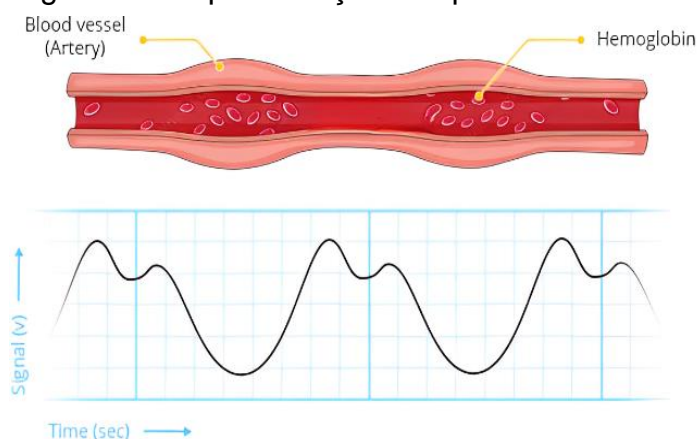


Fonte: adaptado de Last Minute Engineers (2023).

No caso da aferição de batimentos cardíacos, o sensor utiliza os LEDs vermelho e infravermelho para emitir luz através do dedo do paciente. Essa luz é refletida de volta para o sensor através do sangue que flui nas veias do paciente. Ao medir a quantidade de luz refletida, é possível detectar variações causadas pelos batimentos cardíacos (bpm⁹), gerando assim uma forma de onda de sinal que permite determinar a frequência cardíaca. (Figura 6).

Já para medir a quantidade de oxigênio no sangue, funciona de uma forma parecida, a hemoglobina oxigenada (HbO₂¹⁰) é verificada analisando-se a quantidade de luz infravermelha e vermelha absorvida. Quanto mais oxigenado estiver o sangue, mais luz infravermelha será absorvida. Com base nessa absorção de luz, é possível estimar o nível de oxigenação do sangue. A quantidade de hemoglobina oxigenada está relacionada à cor vermelha do sangue. De modo geral, quando calculado a proporção de luz infra vermelho e a luz vermelha recebida pelo fotodetector, o nível de oxigênio (SpO₂) no sangue é definido.

Figura 6 – Representação Frequência Cardíaca.

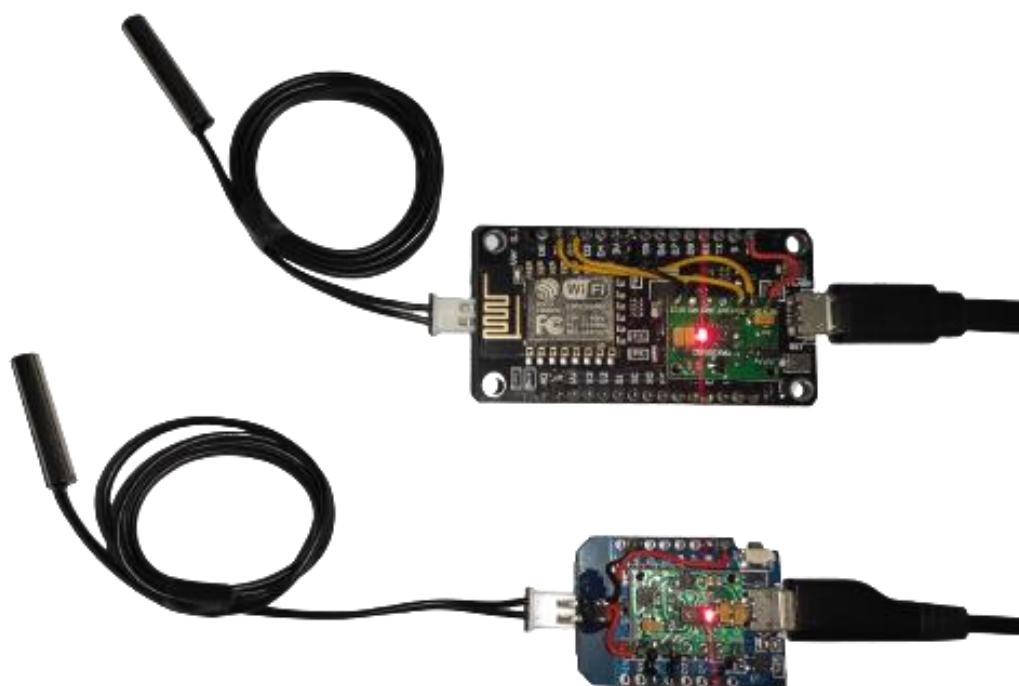


Adaptado de lastminuteengineers.com (2023)

⁹ Batimentos por minuto.

¹⁰ Hemoglobina oxigenada.

Figura 7 – Foto do ESP8266 com os sensores posicionados



Fonte: Acervo dos próprios autores (2023).

A figura 7 apresenta as imagens do dispositivo ESP8266 (tamanho normal e a versão mini, respectivamente) com o sensor MAX30100 e o sensor de temperatura (NTC) posicionados.

Neste capítulo, foram escolhidos componentes eletrônicos para construir um dispositivo de monitoramento de sinais vitais, incluindo o ESP8266 e o sensor MAX30100 para medir batimentos cardíacos e oxigenação do sangue e um sensor de temperatura NTC. O próximo capítulo abordará a construção do *software* para integração do *hardware* e a transmissão dos dados para a plataforma de monitoramento remoto.

4. PROTÓTIPO DO SOFTWARE

O código desenvolvido e que está sendo usado no ESP8266, foi escrito na plataforma Arduino IDE, de modo geral, o código utiliza bibliotecas para a medição dos valores dos sensores, como por exemplo o “MAX30100_PulseOximeter” para a medição da oximetria de pulso/SPO2 e a biblioteca DallasTemperature para a medição da temperatura.

O programa tem uma rede *wifi* previamente configurada, com seu respectivo nome e senha da rede, podendo ser alterada ou adicionadas mais redes (manualmente), além de deixar um ip fixo, para evitar conflitos e também para configurar no código HTML.

Com o *wifi* conectado, o programa é executado no NodeMCU (ESP8266) que atua como servidor HTTP. Ele expõe um endpoint que retorna um JSON contendo os valores

dos batimentos cardíacos, SPO2 (saturação de oxigênio) e temperatura. O servidor é configurado para responder a solicitações HTTP recebidas na porta 80 (podendo ser alterada).

Segundo *Random Nerd Tutorials*

JSON é usado principalmente para serializar e transmitir dados estruturados por conexão de rede – transmitir dados entre um servidor e um cliente. É frequentemente usado em serviços como APIs (*Application Programming Interfaces*) e serviços da *Web* que fornecem dados públicos. (2023, tradução nossa)

Inicialmente, escolheu-se configurar um servidor local para o protótipo, principalmente pela segurança, pois com um servidor sem acesso à internet, há uma camada de segurança adicional, além dos dados não precisam sair do hospital, tendo em vista que é para um controle dos médicos/enfermeiros(às). Porém, caso futuramente necessite ou se deseje instalar um servidor *web*, há possibilidades de implementar esse recurso.

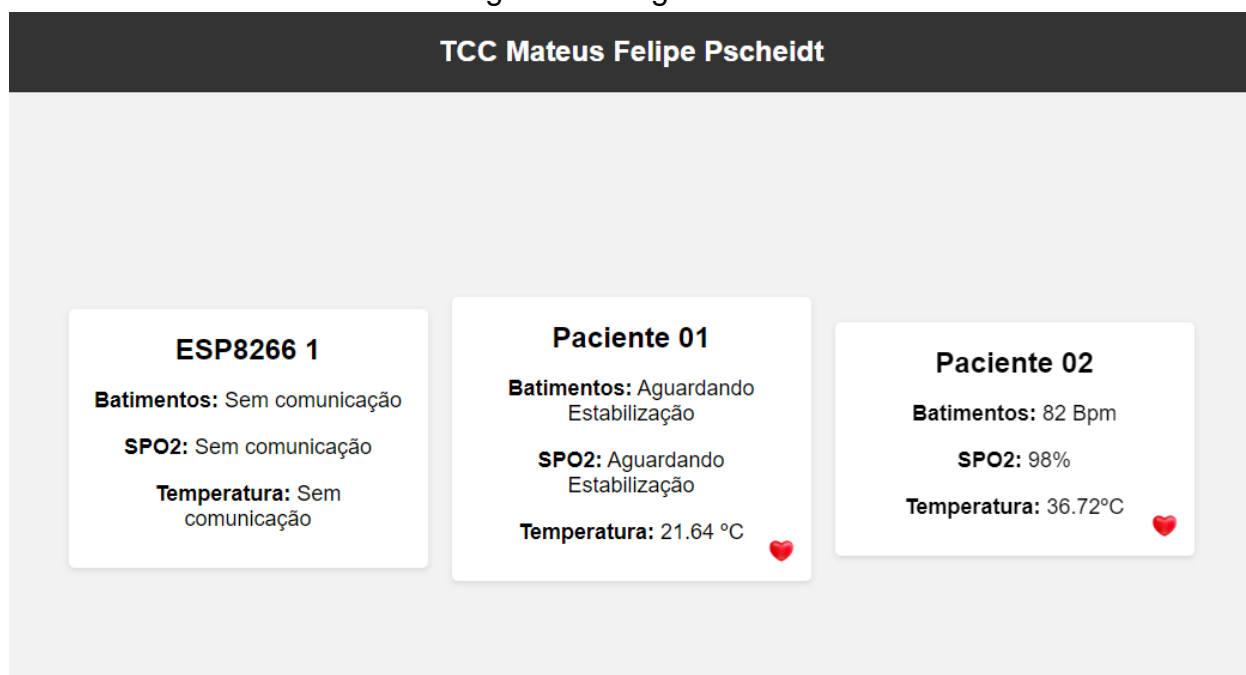
A principal vantagem de utilizar o JSON, é a possibilidade de utilizar mais de um ESP8266 em uma única página *web*, pois sem o JSON seria possível utilizar apenas um dispositivo por página, além de limitar a estilização via CSS e de outras *frameworks*. Com o uso do JSON, o projeto se tornou escalável, pois se pode adicionar inúmeros ESP's na mesma rede, e um servidor (ESP8266) recebendo os valores de todos os sensores simultaneamente.

Já a parte do sistema *web*, é uma página HTML com *JavaScript* que é carregada em um navegador web. Ele utiliza a biblioteca Axios para enviar requisição HTTP do tipo POST para os endereços IP dos servidores NodeMCU (exemplo: "http://192.168.137.174") e receber as respostas em formato JSON. Os valores recebidos são então exibidos na página HTML, sendo eles de batimentos cardíacos, SPO2 e Temperatura. O código busca as informações dos dispositivos a cada 500 milissegundos via JSON (podendo ser alterado).

Portanto, o HTML faz solicitações periódicas aos servidores NodeMCU e faz atualizações dos valores exibidos na página HTML (respostas recebidas). Permitindo assim que as informações coletadas pelos NodeMCUs sejam mostradas dinamicamente na página HTML, possibilitando a visualização dos dados (batimentos cardíacos, SPO2 e temperatura corporal) em tempo real.

É possível utilizar diversos conjuntos de ESP8266s com os respectivos sensores ao mesmo tempo, porém é necessário que todos os ESP8266s estejam na mesma rede *wifi* e com os IP's previamente configurados (na mesma classe de IP).

Figura 8 – Página HTML



Fonte: Acervo dos próprios autores (2023).

A página HTML (Figura 8) apresenta os valores dos sensores de cada ESP8266 (paciente), podendo ser alterado o campo do nome do paciente para um melhor controle, enquanto o sensor MAX30100 não detecta nenhuma leitura, os campos “Batimentos”, “SPO2” e “Temperatura” ficam “Sem comunicação”. Quando a página HTML recebe os valores do ESP8266, e estes não estão estáveis, é exibida a mensagem "Aguardando Estabilização", e quando estabilizado, são exibidas as informações. Cada ESP8266 tem uma “caixa”, e se estiver comunicando corretamente, a imagem de um coração permanece no canto inferior direito de cada caixa.

Neste capítulo, foi descrito o código desenvolvido para o ESP8266, que utiliza bibliotecas para medir os valores dos sensores e para criar um servidor HTTP para lidar com as solicitações e responder com uma mensagem JSON contendo os valores atualizados dos sensores, e a utilização do JSON permitiu a escalabilidade do projeto. A página *web* utiliza o *Framework* Vue.js e a biblioteca Axios para exibir as informações dos microcontroladores. O próximo capítulo apresenta as considerações finais.

5. CONSIDERAÇÕES FINAIS

Após a montagem dos componentes no circuito eletrônico do ESP8266, realizou-se testes individuais das funcionalidades de cada sensor, inicialmente os resultados que foram obtidos com cada sensor foram bem-sucedidos. Seguindo os testes individuais, realizou-se a interligação de todos os sensores simultaneamente, tanto na lógica, assim como na parte do *hardware*. Após diversos ajustes e configurações, foi possível receber os dados de todos os sensores ao mesmo tempo, porém apenas na própria IDE do Arduino, no qual ainda seria necessário realizar a comunicação via rede/HTTP.

Depois de desenvolvido a parte da comunicação via rede e HTTP, realizaram-se diversos testes com todos os sensores bem como a utilização de diversos conjuntos de ESP8266s com os sensores ao mesmo tempo, no qual não se identificou nenhum tipo de problema de comunicação na rede durante os testes.

Afim de testar o *software* com outros tipos de placas, executou-se testes com o ESP8266 e com o ESP8266 mini, qual tem o mesmo chip, porém muito mais compacto. Contudo na versão do ESP8266 mini foi mais complexa a montagem do protótipo, justamente por ser menor, e ter os pontos de conexões próximos uns dos outros.

Durante os testes do sensor de batimentos cardíacos, constatou-se que os valores são alterados se o sensor for exposto a qualquer tipo de luz infravermelho, ou seja, diversas lâmpadas de led convencionais emitem uma pequena quantidade de infravermelho, qual pode alterar o funcionamento do sensor de batimentos cardíacos. Isso ocorre devido ao sensor “ler” a quantidade de infravermelho refletido.

Apesar de ser um protótipo, este projeto se demonstrou viável, pois há um alto potencial para a escalabilidade dentro do sistema hospitalar, bem como a implementação de outros recursos, assim como outros tipos de sensores. De modo geral, os objetivos foram atingidos, uma vez que foi possível desenvolver o protótipo proposto, e realizar as aferições necessárias. Ou seja, este artigo se propôs a desenvolver um protótipo de *hardware* e *software* de um equipamento IoT hospitalar, a fim de facilitar o cotidiano e a vivência hospitalar. Com a finalidade de aplicações em hospitais privados bem como em hospitais públicos, não se limitando a isto, podendo ser implementado em outras áreas, bem como o cuidado de idosos e outras aplicações, por se tratar de um hardware (ESP8266) existente há bastante tempo no mercado, pode-se realizar diversas alterações no projeto.

Contudo, por se tratar de resultados clínicos envolvendo a saúde e bem estar dos indivíduos envolvidos, se faz necessário testes mais aprofundados sobre os sensores apresentados, a fim de verificar a precisão dos dados, assim, possivelmente será necessário a substituição de alguns sensores para outros modelos mais precisos e confiáveis.

Como sugestão para trabalhos futuros, dentre as mais diversas possibilidades, têm-se: a utilização do protocolo de comunicação MQTT, a utilização do protótipo na *internet* (acesso externo), banco de dados (consulta de dados e histórico), botão de emergência integrado ao *hardware*, alarmes específicos entre outros.

REFERÊNCIAS

ALI, Abbas Abd; ALI Adnan Hussein; Al-ASKERY, Ali J. Design and Implementation of Smart E-Health System Based on Cloud Computing to Monitor the Vital Signs. **Real-Time and Measurements Validation**, 2020. Disponível em:

<https://iopscience.iop.org/article/10.1088/1757-899X/745/1/012097/pdf>. Acesso em 19 abr. 2023.

ARDUINO. **What is Arduino?** 2018. Disponível em:

<https://www.arduino.cc/en/Guide/Introduction/>. Acesso em: 21 abr. 2023.

BOULEZHAR, Abdelkader; ABOURICHA, Mostapha; KASMI, Anas El. **A patient's temperature remote control system based on NODEMCU ESP8266**, 2021.

ESP8266. **ESP8266 Introduction**. Disponível em:

<https://fabacademy.org/archives/2015/doc/networking-esp8266.html>. Acesso em 24 abr. 2023.

EMBARCADOS. **Apresentando o módulo ESP8266**. Disponível em:

<https://embarcados.com.br/modulo-esp8266/>. Acesso em 26 abr. 2023.

NODEMCU. **NodeMCU Documentation**. Disponível em:

<https://nodemcu.readthedocs.io/en/dev/>. Acesso em: 21 abr. 2023.

HTML. **Html**. Disponível em <https://webplatform.github.io/docs/html/>. Acesso em 27 abr. 2023.

HTTP. **Http**. Disponível em: <https://devdocs.io/http/>. Acesso em 27 abr. 2023.

LAST MINUTE ENGINEERS. **Interfacing MAX30100 Pulse Oximeter and Heart Rate Sensor with Arduino**. Disponível em: <https://lastminuteengineers.com/max30100-pulse-oximeter-heart-rate-sensor-arduino-tutorial/#fifo-buffer>. Acesso em 22 abr. 2023.

MARTINS, Gabriel de Souza; FERNANDEZ, João Carlos Lopes; PUGLIA, Vinícius de Melo; SANCHEZ, Renato de Brito. **INTERNET DAS COISAS (IOT):**

MONITORAMENTO REMOTO DE SINAIS BIOMÉDICOS. Disponível em:

<https://ojs.eniac.com.br/index.php/Anais/article/view/701/723>. Acesso em 22 abr. 2023.

MQTT. **MQTT: The Standard for IoT Messaging**. Disponível em: <https://mqtt.org/>.

Acesso em: 25 abr. 2023.

OLIVEIRA, Cláudio. **Arduino Descomplicado: Como Elaborar Projetos de Eletrônica**, 2015.

OLIVEIRA, Ricardo Rodrigues. **Uso do Microcontrolador ESP8266 para automação residencial**, 2017. Disponível em:

<http://repositorio.poli.ufrj.br/monografias/monopoli10019583.pdf>. Acesso em 20 abr. 2023.

OPENSOURCE. **What is open source?**. Disponível em:

<https://opensource.com/resources/what-open-source>. Acesso em 30 abr. 2023.

ORACLE. **What is IoT?**. Disponível em: <https://www.oracle.com/internet-of-things/what-is-iot/#:~:text=What%20is%20IoT%3F,and%20systems%20over%20the%20internet>.

Acesso em 30 abr. 2023.

RANDOM NERD TUTORIALS. **Decoding and Encoding JSON with Arduino or**

ESP8266. Disponível em: <https://randomnerdtutorials.com/decoding-and-encoding-json-with-arduino-or-esp8266/>. Acesso em 22 abr. 2023.

RUFINO, Nelson Murilo de O. **Segurança em Redes sem fio**. 2005.

TAREQ, Roaa Wadullah; KHALEE, Turkan Ahmed. **Implementation of MQTT Protocol in Health Care Based on IoT Systems: A Study**, 2021.