

MRP– MONITORAMENTO REMOTO DE PACIENTES EM AMBIENTE HOSPITALAR UTILIZANDO IOT - LORA (DESENVOLVIMENTO DA REDE DE COMUNICAÇÃO)

Filipe Soares Marigo, CEFET-MG

Resumo—Quando em 2020 surgiu a pandemia do novo coronavírus, o medo do espalhamento da doença e lotação de hospitais nos fez ficar em *lockdown*. Com isso tivemos que repensar diversos costumes e protocolos atualmente utilizados. É pensando nessa nova necessidade, que o trabalho deseja criar um sistema de monitoramento onde o paciente usa uma pulseira capaz de realizar o sensoriamento da temperatura, batimentos cardíacos e outras funções vitais em tempo real e comunicar com uma central através do protocolo LoRa. Este trabalho é focado na estrutura de comunicação entre um transceptor e receptor gateway, além do tratamento dos dados para envio com segurança para um banco de dados. Ao final deste trabalho, foi possível analisar a utilização da comunicação LoRa através da eletrônica vestível. O projeto foi desenvolvido por três alunos, Filipe Soares Marigo (atual trabalho), Lucca Defelippo De Paula Romanelli e Vinicius Santana Nasser, que juntos criaram todo o sistema.

Index Terms—Wearables, Tecnologias Vestíveis, IoT, Internet das Coisas, Monitoramento Remoto, Tecnologia na Medicina, LoRa, LoRaWAN.

I. INTRODUÇÃO

O surto da doença causada pelo novo Coronavírus foi caracterizado como pandemia, em 11 de março de 2020 pela Organização Mundial de Saúde (OMS). A superlotação das unidades de saúde, a falta de leitos para internação e de equipamentos para cuidados, são problemas que agravam a situação de atendimento e monitoramento hospitalar [1].

A observação dos indicadores das funções vitais de pacientes, como a frequência respiratória, saturação de oxigênio no sangue, pressão sanguínea, frequência cardíaca e temperatura corporal, é uma das práticas mais comuns e importantes em hospitais [2], uma vez que já é conhecida e estudada a relação entre a piora do quadro clínico do paciente e a alteração nestes indicadores horas antes do quadro de saúde piorar [3]. Portanto, possuir uma forma constante de monitoramento de alguns sinais vitais dos pacientes de forma não invasiva junto

Filipe Soares Marigo, Belo Horizonte, Minas Gerais (filipe.marigo@outlook.com)

Trabalho de Conclusão de Curso submetido em 25 de agosto de 2021 e orientado pelo: Anthony Chiaratti, integrante do Departamento de Eletrônica e Biomédica do Centro Federal de Educação Tecnológica de Minas Gerais e coorientado pelo: Túlio Charles de Oliveira Carvalho, integrante do Departamento de Engenharia Elétrica do Centro Federal de Educação Tecnológica de Minas Gerais

à maneira de observar tais características remotamente, pode ser de grande auxílio no cotidiano hospitalar.

Uma solução atual para o monitoramento dos sinais vitais dos pacientes, aproveitando-se de um dos grandes avanços tecnológicos dos últimos tempos, é a utilização dos “wearables”, que são pedaços de roupas, acessórios ou outras peças vestíveis, compostos por diversos sensores e outros componentes eletrônicos para monitoramento da saúde e aplicação de remédios periodicamente [4].

Algumas empresas de saúde já empregam o monitoramento de pacientes com o uso de wearables, principalmente os relógios inteligentes disponíveis no mercado como o Apple Watch e Samsung Galaxy Watch, como forma de melhorar a qualidade de vida e diminuir os custos. O monitoramento remoto e em tempo real de pacientes em casa auxiliou na diminuição da quantidade de casos em que houve a necessidade de intervenção hospitalar (23% menor que em pacientes que não utilizavam) e de casos de readmissão ao hospital após liberação (44% menor chance de retorno ao hospital em 30 dias e 38% menor em 90 dias), o que acarretou em uma diminuição de 11% [5].

Porém, ainda faltam estudos sistemáticos da aplicação do uso destes wearables em ambientes hospitalares para uma maior gama de idades e quadros clínicos diferentes [6].

A relevância deste trabalho como um todo, se encontra no próprio contexto em que está presente a pandemia, na dificuldade de acesso a este tipo de tecnologia devido a seu custo e na importância do papel que a mesma pode desempenhar em ambientes hospitalares, como visto nos resultados do questionário aplicado nesta pesquisa, visando levantamento de informações. Desta forma, modernizando o sistema de monitoramento da evolução do quadro clínico dos pacientes, podendo assim auxiliar no atendimento preditivo de um paciente que podem necessitar de atenção imediata.

Pretende-se portanto, buscar uma solução barata e acessível através de uma revisão do conteúdo disponível sobre a construção de wearables, verificando evidentemente as exigências hospitalares a serem seguidas pela Norma Brasileira, para construção de uma pulseira de monitoramento de sinais vitais capaz de mostrar os seus dados remotamente e a

exibição dos mesmos em uma interface gráfica.

O trabalho completo, realizado em conjunto com os trabalhos de Vinícius Santana e Lucca Romanelli, busca o desenvolvimento de um sistema para monitoramento remoto de sinais vitais de pacientes em ambiente hospitalar, com base no conceito de Internet das Coisa (IoT), utilizando a tecnologia LoRa (Long Range), o armazenamento em banco de dados e consulta através de uma interface gráfica. Os sinais vitais do paciente deverão ser monitorados através de um dispositivo *wearable*, a ser construído nesta pesquisa. Os dados coletados na pulseira são enviados para um gateway, usando a tecnologia LoRa e armazenados em banco de dados e a devida apresentação de dados através de uma interface gráfica. A Figura 1 demonstra o fluxo simplificado da comunicação.

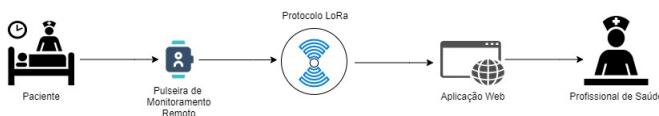


Figura 1. Monitoramento remoto de pacientes em ambiente hospitalar utilizando LORA

Esse trabalho procura realizar a discussão sobre o desenvolvimento da comunicação LoRa e envio dos dados.

II. OBJETIVO

O atual trabalho tem como principal objetivo desenvolver uma rede funcional de comunicação LoRa na frequência de 915 MHz. Em específico, será tratado o protocolo em LoRaWAN os dados entregues pelo hardware da pulseira de monitoramento, como a medição de temperatura, oximetria, frequência cardíaca e posição de um paciente. Além disso, estará presente neste trabalho, o desenvolvimento de uma solução para receber os dados através de um gateway, o tratamento dos dados e o envio para um banco de dados na nuvem sem que haja nenhuma perda das informações, tomando os devidos cuidados de criptografia e autenticação. Com base em revisões bibliográficas, será escolhido um gateway que melhor se encaixe na aplicação dentro de um hospital.

III. METODOLOGIA GERAL

Este trabalho é de natureza aplicada, visando à construção de conhecimentos para a solução de um problema específico e prevendo uma aplicação prática na área da saúde, e foi desenvolvido em duas etapas. A primeira etapa, apresentada em uma monografia, constituiu em um estudo sobre a viabilidade e relevância do projeto e uma revisão bibliográfica dos principais conceitos das tecnologias a serem utilizadas.

Para definir a viabilidade do trabalho e buscando entender as condições atuais de monitoramento de pacientes em hospitais públicos e privados, foi realizada uma pesquisa com os profissionais da área da saúde. Essa pesquisa foi feita por meio de um questionário online elaborado com a ferramenta Google Forms, montado com questões abertas e fechadas, visando buscar informações sobre 3 (três) aspectos chave.

O primeiro aspecto foi sobre a forma de conectividade a internet dentro do ambiente hospitalar público e privado, e

quais dispositivos estão disponíveis para uso coletivo dos profissionais. Possibilitando, com isso a definição das aplicações, interfaces finais e quais dispositivos devem ter compatibilidade com o sistema proposto.

O segundo aspecto chave foi relacionado à forma em que os profissionais da saúde acessam os prontuários e históricos dos pacientes. É importante entender o acesso a essas informações para manter o padrão adotado, fazendo com que a nova tecnologia seja inserida de forma não invasiva e natural.

Por fim, o terceiro e último aspecto foi relacionado a tecnologia de monitoramento utilizando “wearables” com a realidade dos hospitais, ou seja, se é comum tal monitoramento na realidade atual e como a inserção da mesma poderia melhorar nos resultados médicos.

Através dessa pesquisa, foi possível observar que possuir uma forma constante de monitoramento de alguns sinais vitais dos pacientes junto à maneira de observar as características de um paciente remotamente, podem ser de grande auxílio no cotidiano hospitalar.

Definiu-se, então, que o projeto seria constituído por uma pulseira de monitoramento remoto, com as informações protocoladas em LoRaWAN, enviados através do LoRa, para um banco de dados e a construção de uma interface capaz de captar as informações e mostrar para o usuário final.

Com isso no sistema temos as seguintes ações:

Primeiro, o cadastro de pulseiras no sistema, onde é registrada e obtém uma identificação única, depois o cadastro de novos pacientes, onde os dados são recolhidos. Por fim, crie-se uma conexão entre o número de identificação da pulseira e a qual paciente essa pulseira está designada. O processo de cadastro é feito em uma página web, onde seriam informadas as informações necessárias para identificação do paciente.

Segundo, a atualização dos dados dos sensores da pulseira para o banco de dados de forma periódica. Os dados dos sensores usados para monitorar os indicadores das funções vitais são agrupados pelo microcontrolador e enviados, com um formato de mensagem padrão já estabelecido, através do protocolo LoRa.

Terceiro, acesso aos dados através de uma interface de usuário. Ação pela qual o profissional da equipe médica pode acessar os dados de cadastro do paciente assim como as últimas leituras dos sensores relacionadas a aquele paciente. A interface pode ser acessada tanto manualmente quanto através de uma tag NFC colocada na pulseira.

Um diagrama representando esses diferentes métodos de utilização pode ser visto na Figura 2.

A produção do produto final, a Pulseira de Monitoramento Remoto dos pacientes, e o processo de pesquisa necessário será realizado em conjunto com os alunos Vinícius Santana, do trabalho MRP– Monitoramento Remoto de Pacientes em ambiente hospitalar utilizando IOT - LORA (Desenvolvimento do Hardware), e Lucca Romanelli, do trabalho MRP– Monitoramento Remoto de Pacientes em ambiente hospitalar utilizando IOT - LORA (Interface Gráfica).

Além disso, foram levantadas as referências das principais formas de utilização da modulação LoRa, os conceitos, as

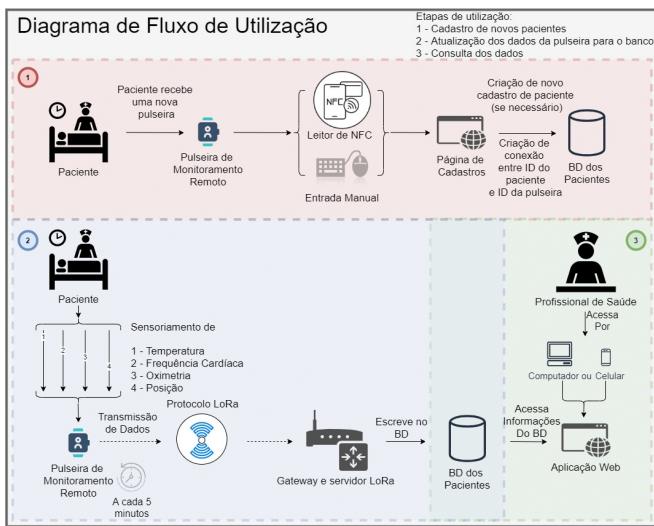


Figura 2. Diagrama da arquitetura do monitoramento de pacientes

suas aplicações e comparações com demais tecnologias de comunicação que podem ser encontradas na monografia entregue durante a primeira etapa do TCC.

IV. METODOLOGIA ESPECÍFICA

O atual trabalho tratará sobre a parte de comunicação do sistema, como o direcionamento das informações recebidas para um banco de dados na nuvem e a validação da utilização de modulação LoRa. A tecnologia LoRa é indicada para a aplicação e por isso foi escolhida como ferramenta de comunicação [7].

A comunicação entre os componentes do sistema é composta por diversas estruturas e linguagem de programa, tornando o projeto versátil e multidisciplinar. A Figura 3 mostra um diagrama que explica visualmente como são as etapas de comunicação entre diferentes pulseiras e os diversos servidores que compõem o sistema.

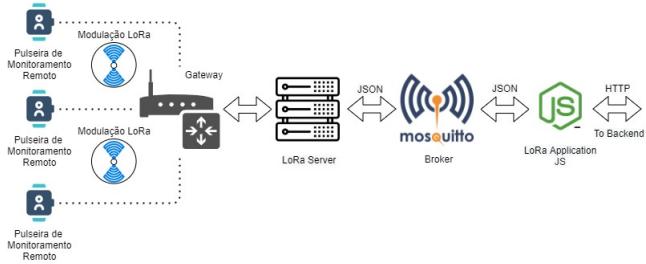


Figura 3. Diagrama simplificado entre as etapas de comunicação

Para melhor entendimento, cada módulo da comunicação e sua aplicação em nosso sistema será discutido separadamente.

A. Interface transceptor e modulação LoRa

O transceptor utilizado para interface LoRaWAN foi o murata CMWX1ZZABZ-078 já descrito na monografia do projeto [11]. Esse dispositivo deve ser programado de acordo com a necessidade de utilização através da linguagem C. Para

a aplicação da pulseira foram usados os principais módulos de *sleep* e *weak-up* através do watchdog, o módulo *USART* para comunicação do transceptor com o microcontrolador. Além disso, cada murata possui seu *device ID* específico que facilitará a identificação de cada pulseira e a associação a cada paciente.

Todas as informações dos pacientes são interpretadas e adequadas por um microcontrolador presente no hardware da pulseira, melhor descrito no projeto do hardware pelo Vinicius Nasser, que envia via comunicação serial para os registradores do murata. As informações principais a serem protocoladas no pacote LoRaWan são:

- Identificador do JavaScript (1 byte);
- Status da pulseira (1 byte);
- Batimento cardíaco do paciente (2 bytes);
- Temperatura do paciente (2 bytes);
- Oximetria do paciente (2 bytes);
- Posição (2 bytes);

A Tabela I mostra a definição de cada código dado pelo byte de status da pulseira.

Tabela I
DEFINIÇÃO DE CADA INFORMAÇÃO DADA PELO REGISTRADOR STATUS

Status	Nome	Dados
1	Power-up	0x00
2	Conectado	0x00
10	Enviando Temp	temp
11	Enviando Freq/Oxi	Freq+Oxi
12	Enviando Pos	x+y+z
13	Enviando Bat	Bateria
14	Enviando Todos	Var sum
20	Bateria Vazia	0x00
21	Bateria Cheia	0x00
22	Bateria Carregando	bateria
30	Ler dado Temp	0x00
31	Ler dado Freq/Oxi	0x00
32	Ler dado Pos	0x00
33	Ler dado Bat	0x00
34	Ler dado Todos	0x00
40	Muda Período Temp	período
41	Muda Período Freq/Oxi	período
42	Muda Período Pos	período
43	Muda Período Bat	período

O pacote contendo as informações (payload) é construído no murata através de uma vetor com os bytes acima respeitando a ordem listada. Todo o pacote é criptografados com base em uma chave conhecida pelo murata e pelo gateway chamada de *app session key* [12], protocolado em LoRaWAN, modulado em LoRa pelo transceptor SX1276 presente dentro do módulo murata e enviado através de uma antena cerâmica para canal de comunicação de 915 MHz. A frequência de envio, fator de espalhamento e demais informações da modulação devem ser configuradas também pelo murata, para nossa aplicação foi usado a banda de frequência australiana AU-915 e o fator de espalhamento igual a 9 foi suficiente para enviar todo o payload sem perda de qualidade do sinal ou aumento do tempo de envio. A Figura 4 mostra um fluxo que resume o processo de modulação do sistema, criptografia dos dados e configuração do LoRa.

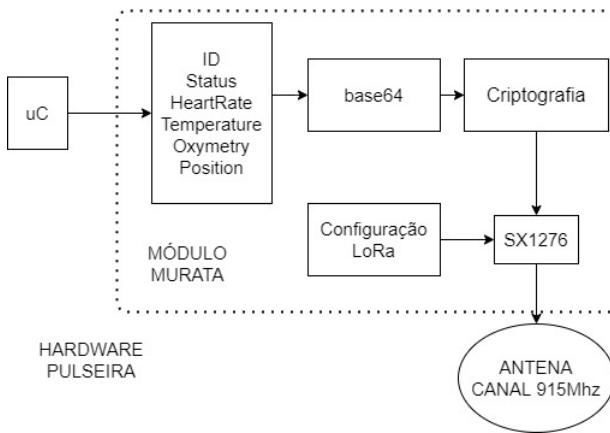


Figura 4. Fluxo das informações no murata, após receber os dados do microcontrolador

B. Serviços atuantes no Gateway

O gateway é um componente delicado de ser escolhido devido ao seu custo e às funcionalidades disponíveis. Foi escolhido o gateway da Multitech Conduit IP67 [13] por três motivos principais: Primeiro, por ele estar equipado com placa que suporta a frequência de 915 MHz, segundo, o gateway é embarcado com sistema operacional mLinux [14], o que facilita a criação de aplicações para o projeto e terceiro, o grupo possui fácil acesso a este modelo gateway para validação do projeto.

O sistema operacional sendo Linux é possível a criação de diversas aplicações para transferência e manipulação dos dados na rede utilizando, por exemplo, o node.js. O fluxograma dado pela Figura 5 mostra os serviços que atuam no gateway e suas respectivas pontes de comunicação.

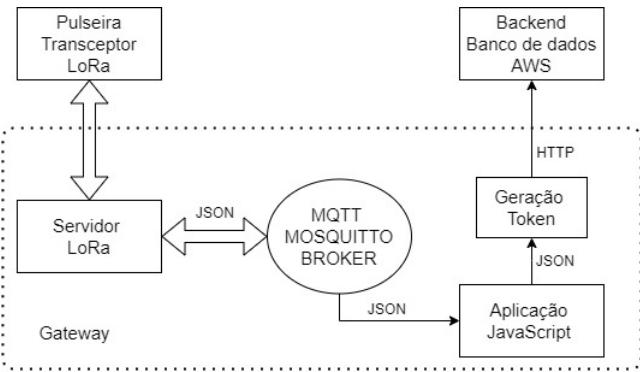


Figura 5. Serviços e pontes que atuam no gateway para adequação e transporte das informações para diferentes aplicações

Para facilitar o entendimento da importância de cada um dos serviços para essa aplicação, será descrito individualmente cada um nos tópicos a seguir mostrando suas características e sua implementação.

1) *Gateway e o servidor LoRa*: Um gateway LoRa pode ser configurado de diversas maneiras. Para essa aplicação, ele foi configurado para a banda de frequência AU-915, assim como

o transceptor. Além disso, para facilitar a aplicação, o modo em que o gateway cadastra novos dispositivos é automático, ou seja, qualquer dispositivo LoRa que esteja dentro da distância limite da antena e que transmite na banda de frequência 915 Mhz será lido por esse gateway, por isso a necessidade de um identificador nos dados transmitidos pela pulseira. O gateway foi conectado a uma rede com acesso a internet via cabo RJ45.

Quando um dispositivo final transmite alguma informação nessa banda de frequência o gateway capta através do servidor LoRa. Esse servidor é o principal responsável pela descriptografia do payload. Ele constrói um JSON, estrutura composta por uma *key* e seu respectivo valor, com várias informações do sinal recebido como por exemplo o device ID, a data e hora do recebimento, a relação sinal-ruído, o payload recebido, a frequência no qual foi comunicado e entre outras informações que podem ser relevantes para a aplicação.

Em seguida, o servidor LoRa publica esse JSON em um tópico MQTT pré-configurado para receber informações vindas de um transceptor, com o nome de “**lora/+/up**” [15]. Nesse momento, nenhuma adequação de dados é realizada, o servidor LoRa tem a função apenas de publicar os dados recebidos em um formato padrão.

2) *MQTT e o broker*: Em paralelo com o servidor LoRa há o serviço MQTT, no qual é um protocolo de comunicação de baixo custo computacional que utiliza o princípio de publicações e inscrições em um determinado tópicos. Esse serviço conta com um *broker* que é um gerenciador de todas essas inscrições, publicações e tópicos. Foi utilizado o broker Eclipse Mosquitto [16] devido a sua popularidade e por ser de código aberto.

Para essa aplicação, o mosquitto tem a função principal de administrar o tópico onde o servidor LoRa publicará os dados e realizar a interface com uma aplicação capaz de enviar os dados para o backend projetado pelo Lucca Romanelli no artigo sobre a interface do projeto com o usuário. Sabendo que inúmeros clientes podem inscrever no tópico onde a mensagem é transmitida, o propósito foi desenvolver um script que fosse capaz de inscrever no tópico MQTT, adequar as informações e enviar para o backend.

O broker é o primeiro contato com os dados JSON vindos da pulseira e a forma como esses dados são descritos [15]. O mosquitto apenas transmite os dados no tópico.

3) *JavaScript para tratamento e envio de dados*: As últimas etapas que compõem o fluxo de dados no gateway são o tratamento dos dados e o envio para o banco de dados. Todo o serviço do banco de dados foi desenvolvido na Amazon Web Services (AWS), por tanto, a melhor forma de se conectar e realizar o envio dos dados seria através do protocolo HTTP.

O método escolhido para adequação e envio dos dados foi o desenvolvimento de um script na linguagem JavaScript juntamente com a criação de uma aplicação em Bash que executa o *script* JS sempre que o gateway for iniciado. Primeiramente, a aplicação JavaScript inscreve no tópico do MQTT onde o servidor LoRa publica os dados. O *script* aguarda uma informação ser publicada, sempre que há uma publicação, os dados do payload presentes no pacote JSON

são convertidos em hexadecimal para que sejam tratados. O primeiro tratamento é feito através do identificador enviado pela pulseira. O primeiro byte convertido deverá conter a informação 0x0E para que seja uma informação válida.

Caso os dados publicados no tópico MQTT sejam válidos, um novo JSON é criado agora com as informações a serem enviadas para o banco de dados em seu formato correto e com as suas respectivas casas decimais. As principais informações escolhidas e filtradas para nossa aplicação podem ser visto através da criação do novo JSON dado pela Figura 6.

```
const JsonDadosPulseira = {
  node: json.deveui,
  time_utc: json.time,
  seqno: json.seqn,
  rssi: rssi,
  snr: snr,
  senStatus: sensor_info.senStatus,
  heart_Rate: sensor_info.heartRate,
  temperature_degC: sensor_info.temp,
  oximetry: sensor_info.oximetry,
  position: sensor_info.position
}
```

Figura 6. JSON construído a partir dos dados filtrados

O JSON é composto pelo node, no qual é a ID do módulo que realizou a comunicação, a key “time-utc” armazena o timestamp do recebimento da informação e ajuda a saber o momento exato em que a informação foi recebida, o seqno é uma sequência numérica que é incrementada no módulo sempre que se realiza um envio. Essa variável permite observar se houve alguma perda de envio de pacote ou se o dispositivo foi reiniciado durante o seu funcionamento, já que o incremento dessa variável é dada de forma contínua e reinicia sempre que o módulo é desligado. O RSSI indica o nível da potência recebida, quanto maior o valor RSSI, maior é a intensidade do sinal e quando medido em números negativos, o número que está mais perto de zero geralmente significa um sinal melhor. O SNR é a relação sinal ruído dado em decibéis e ajuda a entender a qualidade com que a informação chegou no gateway. As demais informações presentes no JSON são as informações úteis da pulseira.

Todos os dados a serem enviados devem apresentar uma autenticação bearer para que a comunicação com o backend via HTTP seja validado. Antes de enviar o novo pacote JSON, um token é gerado com o auxílio da biblioteca “JSON-web-token”. Esse token tem um tempo de expiração que pode ser escolhido de acordo com a necessidade de tempo para o envio da pulseira. Para a aplicação atual o tempo de expiração do token é de 10 segundos, quando o tempo expira, nenhuma ação é permitida. O token gerado é uma informação criptografada que leva em consideração uma chave privada única relacionada com o banco de dados. O token é gerado tanto no gateway quanto no backend e uma comparação entre ambos é o mecanismo que permite um acesso com banco de dados do AWS.

V. RESULTADOS E DISCUSSÕES

Considerando que o microcontrolador presente no hardware da pulseira será capaz de comunicar com o módulo murata via serial e salvar os dados nos registradores, uma forma de testar a modulação do payload em LoRa sem que a pulseira estivesse pronta foi utilizando um kit de desenvolvimento LoRa que contém o módulo do murata. A placa de desenvolvimento *B-L072Z-LRWAN1* [17], mostrada na Figura 7, foi uma placa que permitiu substituir a necessidade da pulseira pronta para os testes iniciais e validação da comunicação entre os diferentes serviços presentes no gateway.



Figura 7. Kit de desenvolvimento LoRa da ST. [17]

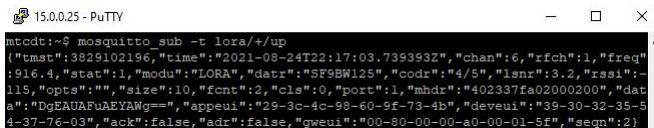
Primeiramente, para validar o funcionamento da modulação e a montagem do payload, foram salvos nos registradores do murata as variáveis com valores constantes, mostrado na Figura 8. Além disso, foi estabelecido que os pacotes de informações seriam enviadas de um em um minuto para o canal de comunicação.

```
uint8_t identificador = 0x0e;
uint8_t stts = 1;
uint16_t heartRate = 80;
uint16_t temperature2 = 366;
uint16_t Oxymetry = 70;
uint16_t position = 90;
```

Figura 8. Variáveis a serem enviadas com valores constantes

As informações da Figura 8 mostram um payload igual a “0E 01 50 16E 46 5A” em hexadecimal. Quando codificado na base64 deverá apresentar um resultado de “DgEAUAFuAEYAWg==”, ou seja, esse valor seria a informação exata recebida pelo gateway quando o payload apresentar esse valores salvos no registrador do murata.

Antes de iniciar a aplicação com o JavaScript, uma forma de verificar se os dados estão chegando corretamente no gateway e sendo publicados de fato no tópico “lora/+up” configurado é inscrever um servidor local neste tópico e observar as interrupções do broker quando há publicações do servidor LoRa presente no gateway. Para as informações constantes enviadas inicialmente e mostrada na Figura 8 podem ser vistas sendo publicadas no tópico mostrada pela Figura 9.



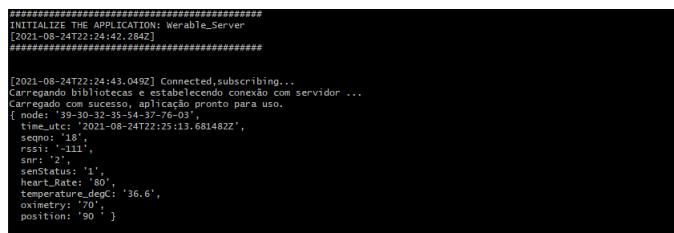
```

15.0.0.25 - Putty
mtcdt:~$ mosquitto_sub -t lora/+up
{"tmst":13829102196,"time":"2021-08-24T22:17:03.7393932","chan":6,"rfch":1,"freq":916.4,"stat":1,"modu":"LORA","datr":"SF9BW125","codr":4/5,"lsrc":3.2,"rssii":115,"opts":{},"size":10,"fcnt":2,"cls":0,"port":1,"mhdr":402337fa02000200,"data":"DgEAUAFuAEYAWg==","appeui":29-3c-4c-98-60-9f-73-4b,"deveui":39-30-32-35-54-37-76-03,"ack":false,"adr":false,"gweui":00-80-00-00-a0-00-01-5f,"seqn":2}
  
```

Figura 9. Dados presente no tópico "lora/+up"do moquitto

Observamos aqui, através da keyword “*datr*”, que os dados ainda estão na base64 e devem ser convertidos antes de serem enviados para o banco de dados. O valor do payload na base64 é o mesmo do calculado previamente e por tanto, podemos validar o envio dos dados e a criptografia da pulseira no canal de comunicação LoRa. Além do payload, podemos observar que há múltiplas informações sendo publicados no tópico e boa parte delas podem ser filtradas e adequadas para cada tipo de aplicação.

Agora que as informações foram validadas observando o recebimento correto pelo gateway e a publicação no tópico MQTT, é possível testar a etapa de tratamento dos dados através do JavaScript. O JavaScript precisou ser projetado de forma minuciosa pois o gateway possui uma limitação de versionamento do node. O node.js atualmente encontra-se na versão 16.7.0 e seu *node packet manager* na versão 7.20.3. A versão disponível no gateway é a 0.10.48 com um npm na versão 2.15.1 e não podem ser atualizadas devido ao modelo do gateway. Portanto, as bibliotecas e estruturas de dados que permitiriam a elaboração de um script melhor estruturado estão limitadas. De qualquer forma, as bibliotecas compatíveis [9] e [10] com essa versão do node funcionaram muito bem e através do console linux, é possível identificar o novo JSON sendo filtrado e gerado corretamente, mostrado na Figura 10.



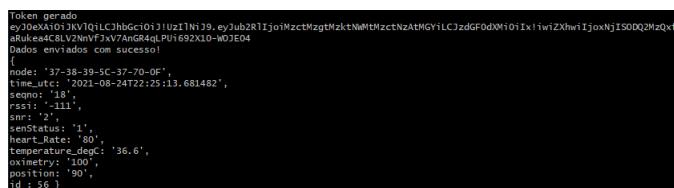
```

INITIALIZE THE APPLICATION: Wearable_Server
[2021-08-24T22:24:42.284Z]
#####
[2021-08-24T22:24:43.049Z] Connected,subscribing...
Carregando bibliotecas e estabelecendo conexão com servidor ...
Carregado com sucesso, aplicação pronto para uso.
{ "node": "37-38-39-6c-37-76-02",
  "time_utc": "2021-08-24T22:25:13.681482Z",
  "seqno": "18",
  "rssii": "-111",
  "snr": "2",
  "senStatus": "1",
  "heart_Rate": "80",
  "temperature_degC": "36.6",
  "oximetry": "70",
  "position": "90" }

  
```

Figura 10. JSON recebido e filtrado pelo JavaScript

Nesse momento, as informações foram adequadas e todos os valores são convertidos para o tipo *string*, de forma a facilitar a manipulação das informações no banco de dados. Em seguida, um token é gerado para autenticação da comunicação via HTTP, a geração do token pode ser vista na Figura 11.



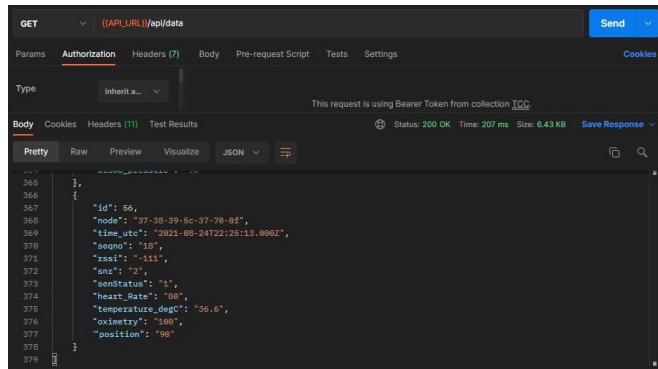
```

Token gerado
eyJOeXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJub2RlIjoiMzctMzgtMzktNmtMzctNzAtMGY1LCJzdGF0dXI0i1xiwiZkhwi1joxNjI5ODQ2MzQxf0
dutkex48L1VNmVfXvAnGr4qPU169ZX10-W0JE04
Dados enviados com sucesso!

node: "37-38-39-6c-37-76-02",
time_utc: "2021-08-24T22:25:13.681482Z",
seqno: "18",
rssii: "-111",
snr: "2",
senStatus: "1",
heart_Rate: "80",
temperature_degC: "36.6",
oximetry: "70",
position: "90",
id: 56
  
```

Figura 11. Geração do Token e envio para banco de dados

Com um token válido e através do método *.post* utilizando a biblioteca *request* do JavaScript é possível enviar as informações filtradas mostradas na Figura 10. Após o envio, os dados devem ser alocados corretamente no banco de dados de acordo com a estrutura do backend. A validação final da comunicação do sistema pode ser feita observando as informações inicialmente enviadas pela pulseira, mostrado pela Figura 8 e alocadas no banco de dados do AWS, mostrado pela Figura 12:



```

{
  "node": "37-38-39-6c-37-76-02",
  "time_utc": "2021-08-24T22:25:13.681482Z",
  "seqno": "18",
  "rssii": "-111",
  "snr": "2",
  "senStatus": "1",
  "heart_Rate": "80",
  "temperature_degC": "36.6",
  "oximetry": "70",
  "position": "90"
}
  
```

Figura 12. Banco de dados analisado através do Postman

O método de tratamento dos dados através do JavaScript, processando dentro do gateway, facilitou consideravelmente na filtragem dos dados a serem salvos no banco de dados e a autenticação bearer trouxe um grau de segurança para o envio pelo protocolo HTTP. Todas as etapas da comunicação se encaixaram sem muitos problemas, mesmo sendo diferentes tipos de serviços.

O gateway até o presente momento não consegue realizar o envio de informações para a placa. A proposta é desenvolver também a capacidade de *downlink* no JavaScript e no gateway de modo que seja possível ajustar configurações dos períodos de aquisição dos dados entre os sensores da pulseira.

VI. CONCLUSÃO

A relevância desta pesquisa em conjunto com as realizadas pelos alunos Lucca Romanelli e Vinícius Santana é de modernizar o sistema de monitoramento de pacientes de forma a obter dados precisos e contínuos, monitorando a evolução do quadro clínico do paciente, podendo assim ter informações relevantes e alertar riscos críticos de um paciente que pode necessitar de atenção imediata. Este trabalho especificamente buscou projetar um meio de comunicação entre o dispositivo final e a interface para o usuário final de forma segura.

Mesmo não sendo possível testar a aplicação no ambiente hospitalar, foi possível identificar, através do sinal-ruído dos dados enviados, que a tecnologia LoRa é promissora para esse sistema e se integrou bem com o transceptor e o receptor gateway selecionados. Além disso, o LoRa suporta longas distâncias dos prédios e tem facilidade de interação com vários dispositivos finais de caráter de baixo consumo energético.

Os trabalhos em conjunto servem como prova de conceito do sistema criado, com o projeto de um protótipo de placa

de circuito impresso capaz de ler os indicadores de funções vitais de pacientes e envia-los utilizando protocolo LoRa; com uma estrutura de servidor central já implantada, que trata dados recebidos pelo servidor LoRa e os adequa ao padrão esperado para armazenamento; e com o desenvolvimento de uma infraestrutura capaz de utilizar os dados reunidos e exibi-los em uma página *mobile-friendly*.

Algumas sugestões podem ser citadas para futuras melhorias no desenvolvimento do projeto. Uma delas é em relação ao gateway da multitech que, mesmo sendo uma ótima opção para o projeto, apresentou algumas instabilidades ao longo dos testes, principalmente devido ao número de aplicações e serviços interagindo simultaneamente. Além disso, a versão do node disponível para esse gateway está defasada e poderia ser facilmente alterada por um sistema mais novo. Essa atualização implicaria em um script de tratamento de dados mais estável, com bibliotecas mais confiáveis de javascript. O gateway até o presente momento não consegue realizar o envio de informações para a placa.

Um outro ponto de melhoria também seria a alteração do módulo transceptor. O murata conseguiu suprir toda a demanda do projeto com facilidade, no entanto, ele possui muitas funcionalidades que não estão sendo utilizadas, ou seja, a troca do módulo murata poderia implicar na redução do tamanho do wearable, tornando-o mais compacto e apresentável.

AGRADECIMENTOS

Eu, Filipe S. Marigo agradeço primeiramente a Deus, pela oportunidade de aprendizagem, o apoio dos familiares, colegas, professores do CEFET-MG e todos aqueles que participaram de alguma forma no desenvolvimento do projeto, em especial o professor orientador Anthony Chiaratti, o coorientador Túlio Charles, os meus pais Claudia Cristina e Jusmar Marigo, os meus colegas Vinicius Santana e Lucca Romanelli pela interação entre os projetos e os meus colegas João Victor Cardoso, Douglas Piva e Jean Castro por todas as ideias e auxílio prestados para a implementação do projeto em geral.

REFERÊNCIAS

- [1] Gallasch, Cristiane Helena et al, “Prevenção relacionada à exposição ocupacional do profissional de saúde no cenário de COVID-19”, Revista Enferm UERJ, Rio de Janeiro, 2020; 28:e49596, DOI: <http://dx.doi.org/10.12957/reuerj.2020.49596> Acessado em: 30 de março de 2021
- [2] Brekke, I. et all, “The value of vital sign trends in predicting and monitoring clinical deterioration: A systematic review”, Publicado: 15 de janeiro de 2019 <https://doi.org/10.1371/journal.pone.0210875> Disponível em: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0210875> Acessado em: 20 de março de 2021.
- [3] Kause, J. et all. “A comparison of antecedents to cardiac arrests, deaths and emergency intensive care admissions in Australia and New Zealand, and the United Kingdom - the ACADEMIA study”. Em: Resuscitation. Disponível em: <https://pubmed.ncbi.nlm.nih.gov/15325446/>. Acessado em: 20 de março de 2021.
- [4] Gouche, Catherine, G. e JONES, J.. “Wearables, wearing, and the rhetorics that attend to them”. Em: Rhetoric Society Quarterly 46(3), pp. 199–206, publicado em 02 de junho de 2016. Disponível em. <https://doi.org/10.1080/02773945.2016.1171689> Acesso em: 20 de março de 2021.
- [5] Maeng, D. et al “Can telemonitoring reduce hospitalization and cost of care? A health plan’s experience in managing patients with heart failure”. Em: Population Health Management 17(6). PMID: 24865986, pp. 340–344., dezembro 2014. DOI: 10.1089/pop.2013.0107. Disponível em: <https://pubmed.ncbi.nlm.nih.gov/24865986/> Acesso em: 20 de março de 2021.
- [6] Lu, Tsung-Chien et al. “Healthcare Applications of Smart Watches. A Systematic Review”. Em: Applied Clinical Informatics 7, pp. 850–869. 14 de setembro de 2016. Disponível em: <https://www.thieme-connect.de/products/ejournals/pdf/10.4338/ACI-2016-03-R-0042.pdf> Acesso em: 20 de março de 2021.
- [7] SEMTECH (2018), “What are LoRa and LoRaWAN?” [Online]. Disponível em: <https://lora-developers.semtech.com/library/tech-papers-and-guides/lora-and-lorawan/>
- [8] LORA ALLIANCE (2015), “A TECHNICAL OVERVIEW OF LORA AND LORAWAN” [Online]. Disponível em: <https://lora-alliance.org/wp-content/uploads/2020/11/what-is-lorawan.pdf>
- [9] Node Packet Manager (Fevereiro, 2020), “Library: Request - Simplified HTTP client” [Online]. Disponível em: <https://www.npmjs.com/package/request>
- [10] Node Packet Manager (2019), “Library: JSON Web Token encode and decode for Node.js version 1.6.3” [Online]. Disponível em: <https://www.npmjs.com/package/json-web-token?activeTab=readme>
- [11] Node Packet Manager (Novembro, 2016), “murata: LoRa Module” [Online]. Disponível em: <https://br.mouser.com/datasheet/2/281/Murata11172016CMWX1ZZABZ-078-1186186.pdf>
- [12] LORA ALLIANCE (2015), “LoRaWAN SECURITY” [Online]. Disponível em: <https://lora-alliance.org/wp-content/uploads/2020/11/lorawansecuritywhitepaper.pdf>
- [13] MultiTech (Julho, 2019), “Conduit IP67 Base Station” [Online]. Disponível em: <http://www.multitech.net/developer/products/multiconnect-conduit-platform/basestation/>
- [14] MultiTech (Julho, 2019), “mLinux” [Online]. Disponível em: <http://www.multitech.net/developer/software/mlinux/>
- [15] MultiTech (Julho, 2019), “MQTT Messages” [Online]. Disponível em: <http://www.multitech.net/developer/software/lora/lora-network-server/mqtt-messages/>
- [16] Eclipse (2019), “Mosquitto: An open source MQTT broker” [Online]. Disponível em: <https://mosquitto.org/>
- [17] STMicroelectronics (Junho, 2019), “STM32L0 Discovery kit LoRa, Sigfox, low-power wireless” [Online]. Disponível em: <https://www.st.com/en/evaluation-tools/b-l072z-lrwan1.html>