

Resumo

[CopyAndPaste] O armazenamento é um dos componentes mais importantes da cadeia de distribuição de vacinas, principalmente pela sensibilidade delas a variações de temperatura, que podem ocasionar diminuição da sua eficácia. Considerando isso, existe uma necessidade de manter constante monitoramento dessa variável, afim de garantir que o produto final venha a manter suas característica originais e a eficiência esperada. Esse trabalho apresenta uma solução baseada em Internet das Coisas, usando comunicação sem fio de baixa potência, para monitorar a qualidade de vacinas, por meio das medidas de temperatura e umidade nos locais de armazenamento. Os dados coletados são organizados em um banco de dados, podendo ser acessados por sistemas decisórios afim de avaliar a qualidade da vacina antes de sua aplicação, evitando problemas em decorrência de problemas de armazenamento.

Palavras-chaves: Monitoramento. Vacinas. IoT. LoRa.

Abstract

[CopyAndPaste] Storage is one of the most important components of the vaccine distribution chain, mainly due to its sensitivity to temperature variations, which can cause a decrease in its effectiveness. Considering this, there is a need to keep constant monitoring of this variable, to guarantee that the final product will maintain its original characteristics and the expected efficiency. This work presents a solution based on the Internet of Things, using low power wireless communication, to monitor the quality of vaccines, by measuring temperature and humidity in storage locations. The collected data are organized in a database, which can be accessed by decision systems to assess the quality of the vaccine before its application, avoiding problems due to storage problems.

Key-words: Monitoring. Vacaciones. IoT. LoRa.

Sumário

1	INTRODUÇÃO	4
1.1	O Programa Nacional de Imunizações	4
1.2	Rede de frio	5
1.3	Justificativa e Relevância do Trabalho	5
1.4	Objetivos	6
1.4.1	Objetivo Geral	6
1.4.2	Objetivos Específicos	6
1.5	Estrutura do Documento	7
2	FUNDAMENTAÇÃO TEÓRICA	8
2.1	Internet das Coisas	8
2.2	Protocolos de comunicação	8
2.2.1	LoRa	9
2.2.2	HTTP/HTTPS	9
2.3	Plataforma de prototipagem	10
2.4	Sensor DHT-22	11
2.5	Servidor	11
2.5.1	Node.js	12
2.5.2	InfluxDB	13
2.5.3	Docker	14
2.6	Aplicativo móvel	15
2.6.1	React Native	15
2.7	Trabalhos Relacionados	16
	REFERÊNCIAS	17

1 Introdução

A saúde é um fator de suma importância para todos os seres vivos, ele é um problema científico, tecnológico, político, prático e filosófico que refere-se a um estado completo de bem estar físico, emocional, social, intelectual e espiritual [1].

Segundo o artigo 196 [2] da Constituição Federal Brasileira a saúde é um direito de todos e dever do Estado garantir medidas políticas sociais e econômicas que visam à diminuição do risco de doenças e de outros agravamentos e ao acesso universal e imparcial às ações e serviços para a sua promoção, proteção e recuperação.

Para garantirmos nossa saúde, precisamos cuidar do nosso corpo e mente, para isto, uma ferramenta que podemos contar são os imunobiológicos, como as vacinas e os soros, diferente de remédios que ajudam no tratamento de pessoas doentes, as imunobiológicos são uma preparação biológica que fornece imunidade total ou parcial de uma determinada doença autoimune para um indivíduo saudável. As vacinas e os soros se diferem pela sua forma de imunização, as vacinas fornece uma imunização ativa, estimulando o nosso organismo na produção de anticorpos, os soros fornecem uma imunização passiva, provendo os anticorpos para o nosso organismo que foram produzidos em outros organismo [3].

Contudo, os imunobiológicos requerem um cuidado elevado para manter a qualidade e sua eficiência, um dos fatores é que são produtos termolábeis, ou seja, se deterioram após determinado tempo expostos a variações de temperaturas e umidade, portanto, é imprescindível assegurar que seu ambiente de armazenagem mantenha uma temperatura e umidade constante [4] para garantir uma longevidade maior para o produto. Para este propósito, existem a Rede de Frio, um processo desenvolvido pelo Programa Nacional de Imunizações, PNI, de conversação, armazenamento e transporte dos medicamentos, objetivando as condições adequadas dos mesmos, mantendo suas características iniciais [4].

No ano de 2014, foi relatado no estudo [5] que a qualidade de conservação das vacinas não eram adequadas em boa parte dos municípios da macrorregião Oeste de Minas Gerais, alguns dos motivos citados foram a má gestão dos refrigeradores, falhas no monitoramento da temperatura e insuficiência de recursos humanos.

1.1 O Programa Nacional de Imunizações

Com o sucesso da Campanha de Erradicação da Varíola, CEV, iniciada em 1965, tendo seu fim em 1973 [6], ampliou dentro do Ministério da Saúde maiores investimento no controle de doenças autoimune, dando um impulso na criação do PNI [7]. O PNI foi fundado com objetivo de controlar e erradicar as doenças imunopreveníveis, através de

ações metalizadas de vacinação da população. Em 1980 foi realizada a primeira campanha de vacinação da poliomielite e desde então foram realizadas diversas campanhas, tais como a da rubéola, sarampo, tuberculose febre amarela [7, 4] e atualmente contra a COVID-19.

De acordo com a Lei n.º 6.259 de 30 de outubro de 1975, regularizada pelo Decreto nº 78.231 em 1976, certificar o PNI, sobre a responsabilidade do Ministério da Saúde e define as seguintes competências [4]:

- implantar e implementar as ações do Programa, relacionadas com as vacinações de caráter obrigatório;
- estabelecer critérios e prestar apoio técnico e financeiro à elaboração, implantação e implementação dos programas de vacinação a cargo das secretarias de saúde das unidades federadas;
- estabelecer normas básicas para a execução das vacinações;
- supervisionar, controlar e avaliar a execução das vacinações no território nacional, principalmente o desempenho dos órgãos das Secretarias de Saúde, encarregados dos programas de vacinação.

1.2 Rede de frio

A Rede de Frio, também chamado de Cadeia de Frio é um processo definido pelo PNI designado a auxiliar os profissionais da área da saúde, responsáveis pela imunização no Brasil para que possa assim, garantir a efetividade e durabilidade dos imunobiológicos e medicamentos termolábeis.

No Manual de Rede de Frio [4] é definido os requisitos dos ambientes de armazenagem para garantir a efetividade dos produtos, desde os laboratórios produtor as instâncias locais, passando pela instância nacional, estadual, e no transporte entre eles. Para as Câmaras frigoríficas, a temperatura de operação é entre -20°C a $+2^{\circ}\text{C}$, variando conforme o material armazenado, para a maioria dos imunobiológicos o recomendado é de $+2^{\circ}\text{C}$ e $+8^{\circ}\text{C}$ para ter um melhor controle da sua validade, havendo algumas exceções, como por exemplo, as vacinas Pfizer-BioNTech e Moderna, produzidas para combater o COVID-19, que precisam ser armazenadas entre -80°C a -60°C e -25°C e -15°C , respectivamente [8].

1.3 Justificativa e Relevância do Trabalho

Atualmente, com o início da distribuição das vacinas contra o COVID-19 em todo o mundo, uma das dificuldades enfrentadas é o controle de qualidade no armazenamento tanto em transporte [9], quanto no local da aplicação justamente por serem produtos

sensíveis a temperatura e necessitem de muita cautela, em contrapartida, por ser um produto com uma demanda elevada, a sua oferta deve ser rápida para que haja imunização em massa da população, encaminhando-se para o fim da pandemia.

Esses desafios enfrentados na distribuição das vacinas do COVID-19 não são exclusivamente deles, enfermeiras de Minas Gerais, com o objetivo de inteirar-se acerca do sistema de manutenção dos produtos, realizaram um estudo sobre a conservação de vacinas em unidades básicas de saúde, UBSs. Nessa pesquisa foi relatado diversas irregularidades no armazenamento dos materiais termolábeis sem o comprimento das normais da PNI, como por exemplo, a presença de vacinas que deveriam ter sido descartadas por terem atingido seu tempo máximo de diluição, ainda presentes nos refrigeradores, cerca de 52% dos imunobiológicos armazenados nos refrigeradores eram estabelecidos erroneamente e 36% dos refrigeradores observados contavam com objetos portas como fracos vazios.

Analisando as temperaturas dos ambientes de armazenagem, foi observado que 4% das unidades não realizavam o registro da temperatura dos refrigeradores, 88% dos refrigeradores usavam termômetros analógicos de baixa confiabilidade e cerca de 12% foram analisados na hora da visita temperaturas abaixo da faixa recomendada ($+2^{\circ}\text{C}$ a $+8^{\circ}\text{C}$), chegando a 0°C , outros estudos realizado [5, 10, 11] também testemunharam essa irregularidade na temperatura em seus respectivos locais ao longo do mundo, valendo salientar o estudo realizado na Bolívia [10], que teve resultados ainda piores, sendo registrado um temperatura mínima de -7.2°C e uma máxima de 22.7°C .

Pensando nesse cenário, esse trabalho apresenta uma alternativa para melhorar a forma de monitoramento de temperatura e umidade realizadas de produtos imunobiológicos por laboratórios, unidades públicas de saúde e afins no intuito de auxiliá-los a manter os materiais em suas melhores condições.

1.4 Objetivos

1.4.1 Objetivo Geral

Construir uma arquitetura baseada em conceitos de IoT visando o monitoramento de temperatura e umidade de imunobiológicos para auxiliar funcionários da saúde, garantindo melhores condições para a vacinação da população frente a incidência de doenças.

1.4.2 Objetivos Específicos

- Construir um protótipo inicial para coleta da temperatura e umidade nos ambientes de armazenagens dos imunobiológicos.
- Implementar um servidor para a armazenagem dos dados coletados e posteriormente

fornecer históricos das temperaturas e umidade ao aplicativo móvel.

- Desenvolver um aplicativo móvel para fornecer uma interface amigável para os usuários auxiliando no controle de qualidade dos produtos.
- Realizar testes e análises dos dados de transmissões a fim de garantir a confiabilidade das temperaturas e umidade coletadas.

1.5 Estrutura do Documento

Os capítulos subsequentes estão organizados da seguinte maneira:

[HM] A ser feito quando o documento tiver pronto

2 Fundamentação Teórica

Neste capítulo será abordado sobre os principais conceitos e tecnologias utilizadas no desenvolvimento deste projeto, do hardware à aplicação móvel, iniciando pelo conceito principal, a Internet das Coisas.

2.1 Internet das Coisas

Com o intuito de ampliar a internet atual, interligando os objetos do nosso cotidiano, animais e humanos em uma única rede, foi criado a Internet das Coisas, IoT, também conhecida como internet de todas as coisas. Para tal, os objetos viram objetos inteligentes, possuindo capacidade de comunicação associados com sensores que fornecem dados para outros dispositivos. Estes objetos, conectados a IoT, se comunicam via internet, trocando dados em tempo real, transmitindo informações acerca do ambiente que estão inseridos e/ou até mesmo dos seus respectivos estados.

Deste modo, é adicionando uma nova gama de possibilidades, trazendo grandes benefícios para ambientes domésticos, mas principalmente para a zona industrial. No primeiro caso, aplicações, tais como: aprendizagem reforçada, monitoramento e vigilância inteligentes e vida assistida têm despontado entre aquelas que mais chamam atenção, tanto dos usuários, como das empresas de desenvolvimento de soluções tecnológicas. No segundo caso, IoT se apresenta como um diferencial competitivo importante em campos tais como automação e manufatura industrial, logística, gestão de processos de negócio, etc.

Do ponto de vista da produtividade, IoT apresenta-se como um importante meio pelo qual pode-se desenvolver aplicações sofisticadas que podem integrar o mundo real e o mundo virtual. Em empresas de manufatura, produtos conectados permitem a existência de um ambiente de serviços e produtos no qual a manutenção pode ser realizada baseada na necessidade real, em vez de uma suposição estatística. Adicionalmente, produtos e máquinas conectados podem receber atualizações de software quando disponíveis, para garantir que estejam sempre funcionando em eficiência ótima.

2.2 Protocolos de comunicação

Protocolos de comunicação são conjuntos de normas que definem uma forma de troca de mensagens entre dois ou mais dispositivos em um determinado meio físico, os meios mais comuns são por cabos de materiais condutores e pelo ar, transmitindo ondas eletromagnéticas, na IoT o que se predomina é por meio sem fio (*wireless*). Neste projeto,

foram utilizados dois protocolos de comunicações, o LoRa e o HTTP.

2.2.1 LoRa

A tecnologia Long Range, LoRa, é uma forma de comunicação sem fio, semelhante ao Wi-Fi e ao Bluetooth, que permite um longo alcance de comunicação com baixo custo. O raio de comunicação sem fio utilizando o LoRa, dependendo do dispositivo selecionado, pode alcançar quilômetros de distância.

Embora o LoRa tenha sido fundamentalmente desenvolvido pela Semtech Corporation, o padrão imposto pelo LoRa permitiu que muitas empresas utilizassem para diversos projetos com um custo benefício satisfatório, aumentando o ecossistema e ganhando um envolvimento significativamente maior, uma variedade maior de produtos e um aumento geral no uso e aceitação.

O LoRa em si diz respeito à camada física, sendo que a camada lógica é chamada de LoRaWAN, que é um protocolo usado pelo LoRa para comunicação entre pontos de conexões de um nós final (*End Nodes*) para envios de informações diretamente a um concentrador (*Gateway*), que centralizar as informações e envia a um determinado sistema [12]. Os End Nodes são dispositivos responsáveis por coletar os dados necessários e transmiti-los a um determinado Gateway, o Gateway por sua vez, é responsável por receber as informações de múltiplos End Nodes e repassá-lo a um determinado sistema, como por exemplo um servidor onde será armazenado esses dados, como podemos ver na figura 1.

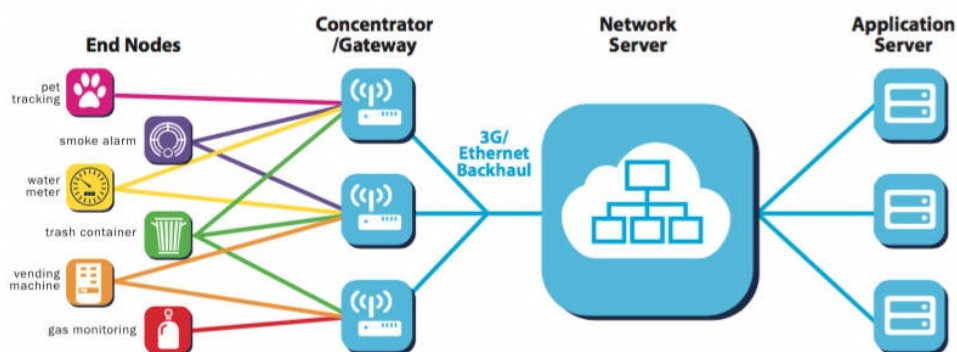


Figura 1 – Representação da relação entre os dispositivos LoRa (Adaptada de [13]).

2.2.2 HTTP/HTTPS

Inicialmente utilizado pelo *World-Wide Web*, WWW, em 1990, como o protocolo base da internet por ser uma protocolo leve e rápido para transições de informações em sistemas distribuídos, o *Hypertext Transfer Protocol*, HTTP, é um protocolo que tramite documentos de diversos formatos através da sistemas distribuídos, mais conhecido pela sua

disseminação pela internet [14]. Quanto anos depois, em 1994, a Netscape Communications criou o HTTPS, uma implementação do HTTP adicionando uma nova camada, a de segurança, utilizando-se do protocolo SSL/TLS, um protocolo que fornece segurança entre comunicações sobre rede de computadores.

O HTTP funcionando baseado no paradigma de cliente e servidor, onde o cliente estabelece uma conexão fazendo uma requisição solicitando algo ao servidor, e o servidor por sua vez, o responde enviando uma mensagem. Tais requisições possuem um método atrelado a ele, esses métodos são usados semanticamente, com o objetivo de organizar e dividir suas responsabilidades, os métodos mais utilizados são: GET, POST, PUT e DELETE. o GET é usado quando o cliente solicita dados ao servidor, o POST quando o cliente quer enviar alguma nova informação, o PUT é semelhante ao POST, entretanto é referente a atualização de um dados existente e o DELETE é uma requisição pedindo a remoção de uma determinada informação [15].

Essas mensagens que transitam entre clientes e servidores, são compostas por linhas de textos divididas em três blocos como pode ser visto na figura 2, o primeiro bloco consiste em uma linha com o método da requisição, a url, ou seja, o caminho do documento seguido pela a versão do protocolo; o segundo bloco são referentes ao cabeçalho HTTP, responsável por fornecer informações e metadados ao servidor, como o tipo de dado transitado ou dados que alteram seu comportamento; o terceiro bloco é respectivo ao bloco de dados, esse bloco é opcional dependendo do tipo de requisição realizada, mais utilizado pelos métodos POST e PUT.

```
POST /contact_form.php HTTP/1.1
Host: developer.mozilla.org
Content-Length: 64
Content-Type: application/x-www-form-urlencoded

name=Joe%20User&request=Send%20me%20one%20of%20your%20catalogue
```

Figura 2 – Exemplo de uma requisição POST (Retirado de [15]).

2.3 Plataforma de prototipagem

Plataformas de prototipagem eletrônica são formas facilitadas de desenvolver um protótipo inicial de determinado projeto. Uma das plataformas de prototipagem existentes mais famosa é o Arduino, uma plataforma de código aberto criada em 2005 na Itália pelo professor Massimo Banzi, que tinha como objetivo, ensinar eletrônica e programação para seus alunos, provendo uma forma acessível e de fácil uso. O Arduino possui seu

próprio ambiente de desenvolvimento de mesmo nome, utilizando-se de sua linguagem de programação baseada em C.

2.4 Sensor DHT-22

O sensor DHT-22 é um sensor de temperatura e umidade da família de sensores DHT comumente utilizado em aplicações da IoT por ser um sensor pequeno (como podemos ver na figura 3) e possuir um baixo custo. Ele opera na faixa de 3.3 a 6 Volts e consegue capturar dados de temperatura entre -40°C a 80°C e umidade entre 0% a 100% RH, com uma acurácia de 0.5°C e 2% RH, respectivamente [16].

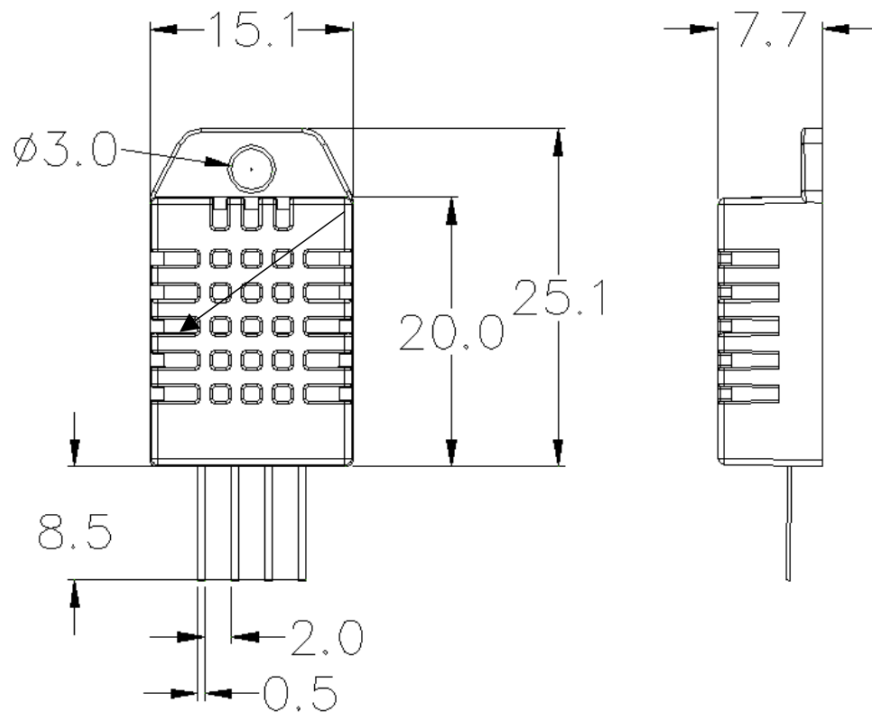


Figura 3 – Dimensões em milímetros do sensor DHT-22 (Retirado de [16]).

2.5 Servidor

Um servidor é um sistema de computação centralizada, responsável por fornecer serviços em uma rede de computadores, tais serviços variam conforme a necessidade do sistema, podendo ser um controlador de domínio, provedor de arquivos, de impressão, de e-mails entre outros. No modelo cliente/servidor (figura 4), por sua vez, o servidor é responsável por receber as requisições de clientes, provendo informações e dados de acordo com a demanda.

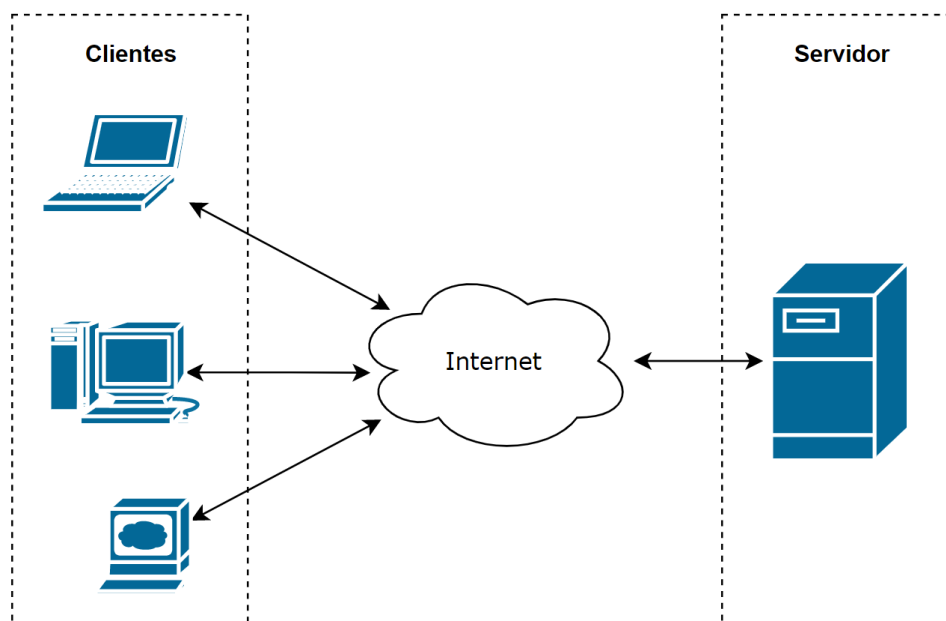


Figura 4 – Representação do modelo cliente/servidor (Autoral).

2.5.1 Node.js

Em 2009, Ryan Dahl criou o Node.js ou simplesmente Node, um ambiente de código aberto de execução Javascript para ser utilizado por servidores (*server-side*) baseado no interpretador V8 da Google. O objetivo do Node é fornecer uma forma de criar serviços com alta capacidade de escala, consumindo pouca memória e de fácil aprendizado. Apesar da linguagem de programação Javascript fornecer uma performance inferior as linguagens já existentes para esse objetivo, seu uso oferece dois grande benefícios, ser uma linguagem de fácil aprendizado e de grande uso por ser a linguagem padrão dos navegadores, e consequentemente do desenvolvimento web [17].

A forma de execução no Node é por eventos assíncronos, diferente da maioria dos outros ambientes modernos que utilizam threads do sistema operacional, SO. O Node usa apenas uma thread assíncrona principal que executa operações de entrada e saída de dados, E/S, chamada de *Event Loop* [18]. O *Event Loop* funciona em um ciclo, escutando uma lista de chamadas, onde são armazenadas as requisições recebidas, e as direcionando cada uma para uma Thread a parte, que executará a função recebida, como podemos ver na figura 5. A priori, o Node possui quatro Threads trabalhadoras, responsáveis por executar as funções, entretanto é possível configurar dependendo da máquina onde está sendo efetuado a instância do Node.

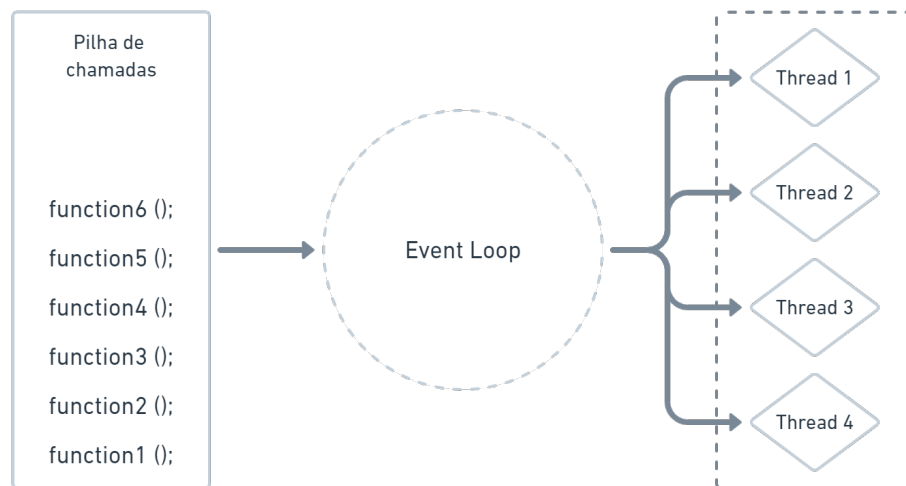


Figura 5 – Ciclo de eventos no Node (Autoral).

2.5.2 InfluxDB

O InfluxDB é um banco de dados que armazena series temporais (*data series database*), ou seja, sua chave é o tempo e sua forma de armazenagem de dados é em ordem cronológica. Ele foi projetado para lidar com grandes cargas de escrita e consulta, perfeita para armazenar dados em tempo real, como monitoramento de DevOps, métricas de aplicativos, big data e dados de sensores da IoT [19]. De forma geral, séries temporais acabam se tornando gráficos em função do tempo em um determinado período, por exemplo a temperatura de um freezer ao decorrer do dia, podendo assim ver facilmente a máxima, a mínima e suas variações. Esses dados podem ser também coletados e feito uma análise mais complexa usando qualquer ferramenta estatística, dependendo da sua necessidade.

O InfluxDB é composto por *databases* (bancos de dados), *measurements* (medições), *fields* (campos) e *tags*. Podemos representar essa estrutura como conjuntos como podemos ver na figura 6, no InfluxDB é possível ter inúmeros *databases*, onde cada *database* contém suas *measurements* que são tabelas de dados correspondente a algum dado em específico, por exemplo, se tivermos 2 sensores que coletam dados diferentes, cada sensor viraria um *measurement*, e cada *measurement* é composto de dois tipo de atributos, os *fields*, onde ficam os dados da sua medida, e as *tags*, que são campos de dados que diferem *fields* por serem campos indexáveis, feitos exclusivamente para realizar buscas, por exemplo, é comum adicionar uma *tag* que seja um identificador do dispositivo que coletou esse medida.

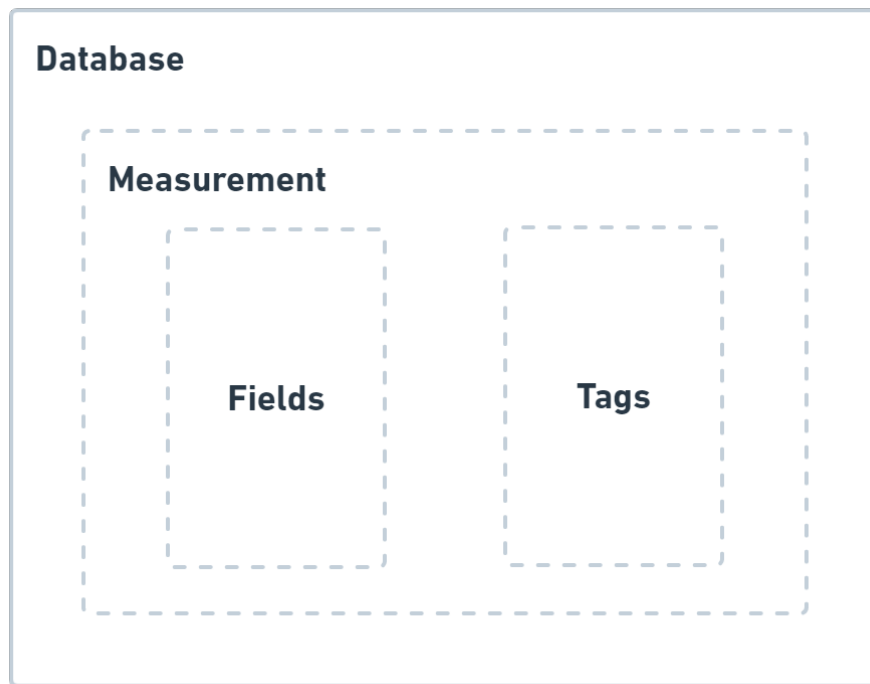


Figura 6 – Composição de um banco de dados do InfluxDB (Autoral).

2.5.3 Docker

Docker é uma plataforma *open source*, desenvolvida utilizando a linguagem de programação GO pela Google com o objetivo de criar facilmente ambientes isolados (containers) com um alto desempenho e que sejam portáteis, sendo uma opção em relação as virtualizações. Desta maneira, é possível, por exemplo, criar inúmeras aplicações usando as mesmas tecnologias, cada uma em um *container* diferente e nenhuma vai interferir na outra, e todas na mesma máquina, e se for preciso replicar em outra máquina, é possível criar uma imagem do container e instalar o mesmo ambiente nesta outra máquina.

Em comparação com as virtualizações, os containers não precisam de um sistema operacional, apenas o essencial para executar determinada função. Dessa forma, os containers conseguem ter um controle maior, consomem menos recursos, ganham uma maior flexibilidade e uma manutenibilidade. Podemos ver a comparação entre virtualização e containers na figura abaixo.

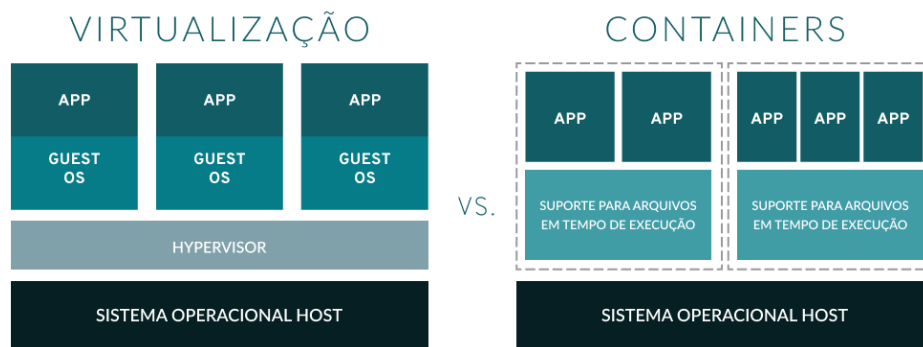


Figura 7 – Comparação entre o modelo de virtualização e modelo de containers (Adaptada de [20]).

2.6 Aplicativo móvel

Aplicativo móvel, APP, é um *software* desenvolvido especificamente para dispositivos móveis, como telefones celulares, tablets, relógios inteligentes entre outros. Cada SO utiliza uma determinada linguagem de programação para desenvolvedores poderem criar os apps. Com uma grande diversidade de sistemas operacionais, com uma linguagem de programação distintas, transformam o desenvolvimento de aplicativos multiplataformas mais trabalhoso e repetitivo, pois tem que ser programada uma versão diferente para cada plataforma. Tendo esse problema em mente, várias formas de programação híbridas foram criadas, com o princípio de poder programar utilizando apenas uma código para várias plataformas distintas, dentre elas, as duas que mais se destacam atualmente são o Flutter e o React Native.

2.6.1 React Native

O React Native é um *framework* para desenvolvimento nativo voltado para dispositivos para os dois principais SO móveis atualmente, Android e iOS. Ele é de código aberto, mantido pelo Facebook e sua comunidade. Ele é um *framework* bastante popular, utilizado por grandes empresas globais atualmente, como Netflix, Uber, a própria Facebook e empresas brasileiras como o EBANX e a Globo [21].

O React Native utiliza a linguagem JavaScript como principal e a renderiza para código nativo, para isso é adicionado uma camada em JavaScript, chamada de *Bridge* que se comunica com o sistema operacional, mandando comandos para renderizar os componentes nativos [22].

2.7 Trabalhos Relacionados

Referências

- 1 FILHO, N. d. A. *O que é saúde?* [S.l.]: Editora Fiocruz, 2011. Citado na página 4.
- 2 MOURA, E. S. D. O direito à saúde na constituição federal de 1988. *Âmbito Jurídico*, XVI, v. 114, 2013. Citado na página 4.
- 3 SOMA, C. *Como funciona o tratamento com imunobiológicos?* 2018. Disponível em: <<https://clnicasoma.com.br/como-funciona-o-tratamento-com-imunobiologicos>>. Citado na página 4.
- 4 SAÚDE, M. da. *Manual de Rede de Frio*. 2001. Disponível em: <http://bvsms.saude.gov.br/bvs/publicacoes/manual_rede_frio.pdf>. Citado 2 vezes nas páginas 4 e 5.
- 5 OLIVEIRA, V. C. d. et al. Avaliação da qualidade de conservação de vacinas na atenção primária à saúde. *Ciência & Saúde Coletiva*, SciELO Public Health, v. 19, p. 3889–3898, 2014. Citado 2 vezes nas páginas 4 e 6.
- 6 MUNIZ, É. S. *Memórias da erradicação da varíola*. [S.l.]: SciELO Brasil, 2011. Citado na página 4.
- 7 TEMPORÃO, J. G. O programa nacional de imunizações (pni): origens e desenvolvimento. *História, ciências, saúde-manguinhos*, SciELO Brasil, v. 10, p. 601–617, 2003. Citado 2 vezes nas páginas 4 e 5.
- 8 NIFORATOS, J. D. Common questions about the pfizer-biontech and moderna covid-19 vaccines. Citado na página 5.
- 9 BAE, J. et al. Challenges in equitable covid-19 vaccine distribution: A roadmap for digital technology solutions. Citado na página 5.
- 10 NELSON, C. et al. Monitoring temperatures in the vaccine cold chain in bolivia. *Vaccine*, Elsevier, v. 25, n. 3, p. 433–437, 2007. Citado na página 6.
- 11 FALCÓN, V. C. et al. A vaccine cold chain temperature monitoring study in the united mexican states. *Vaccine*, Elsevier, v. 38, n. 33, p. 5202–5211, 2020. Citado na página 6.
- 12 ALLIANCE, L. *What is LoRaWAN® Specification*. 2020. Disponível em: <<https://lora-alliance.org/about-lorawan>>. Citado na página 9.
- 13 3GLTEINFO. *LoRa Architecture*. 2020. Disponível em: <<https://www.3glteinfo.com/lora/lora-architecture>>. Citado na página 9.
- 14 BERNERS-LEE, T.; FIELDING, R.; FRYSTYK, H. *Hypertext transfer protocol-HTTP/1.0*. [S.l.]: May, 1996. Citado na página 10.
- 15 MOZILLA. *Uma típica sessão HTTP*. 2019. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Session>>. Citado na página 10.

-
- 16 ELECTRONICS, A. *DHT22 Datasheet*. 2020. Disponível em: <<https://pdf1.alldatasheet.com/datasheet-pdf/view/1132459/ETC2/DHT22.html>>. Citado na página 11.
- 17 TILKOV, S.; VINOSKI, S. Node. js: Using javascript to build high-performance network programs. *IEEE Internet Computing*, IEEE, v. 14, n. 6, p. 80–83, 2010. Citado na página 12.
- 18 FOUNDATION, O. *Documentação oficial do Node.js*. 2019. Disponível em: <<https://nodejs.org/pt-br/about/>>. Citado na página 12.
- 19 GIACOBBE, M. et al. An implementation of influxdb for monitoring and analytics in distributed iot environments. In: SPRINGER. *International conference on the Sciences of Electronics, Technologies of Information and Telecommunications*. [S.l.], 2018. p. 155–162. Citado na página 13.
- 20 HAT, R. *Containers x máquinas virtuais*. 2020. Disponível em: <<https://www.redhat.com/pt-br/topics/containers/containers-vs-vms>>. Citado na página 15.
- 21 BRASIL, R. *Empresas que utilizam React no Brasil*. 2020. Disponível em: <<https://github.com/react-brasil/empresas-que-usam-react-no-brasil>>. Citado na página 15.
- 22 FACEBOOK. *Documentação oficial do React native*. 2020. Disponível em: <<https://reactnative.dev/>>. Citado na página 15.