

# Dashboard IoT Remote Lab with MQTT Protocol

Victor Takashi Hayashi  
victor.hayashi@usp.br  
Universidade de São Paulo  
São Paulo, Brasil

Felipe Valencia de Almeida  
felipe.valencia.almeida@usp.br  
Universidade de São Paulo  
São Paulo, Brasil

Lucas Yuji Harada  
lucas.harada@usp.br  
Universidade de São Paulo  
São Paulo, Brasil

Fabio Hirotsugu Hayashi  
fabio.hayashi@ufabc.edu.br  
Universidade Federal do ABC  
Santo André, Brasil

Reginaldo Arakaki  
reg@usp.br  
Universidade de São Paulo  
São Paulo, Brasil

## ABSTRACT

The Internet of Things enables the data collection of the physical world in real time, raising opportunities for cost reduction and optimization at large scale. However, creating an IoT solution that transforms these data into information by enabling new visualizations and interactions is not trivial. A remote digital electronics lab tool based on the MQTT protocol with energy monitoring, device control and integration with web and mobile dashboards is presented to support remote learning. The open source tool might be useful for teachers and technical staff of digital electronics lab courses by automating the remote lab devices management.

## KEYWORDS

IoT, Cloud, Dashboard, MQTT, Remote Lab

## 1 INTRODUÇÃO

O termo Internet das Coisas (IoT) tem sua origem em uma apresentação de Kevin Ashton para a empresa P&G em 1999 [1]. A Internet ainda é muito dependente de seres humanos para a criação de dados, o que pode fazer com que os sistemas de informação saibam mais sobre ideias do que o mundo físico em que as pessoas vivem. Na visão de Ashton, a Internet das Coisas pode reduzir custos e desperdícios por meio de dispositivos que podem sentir o mundo físico de forma autônoma, sem as limitações de dados inseridos de forma manual por humanos.

Devido ao seu grande impacto econômico e atuando como vetor da transformação digital, observa-se um aumento exponencial de soluções IoT para suportar o controle e monitoramento de forma remota, com estimativa de 26,4 bilhões de dispositivos em 2026 [4].

Por outro lado, os dados coletados pelos dispositivos de Internet das Coisas devem se transformar em informações úteis para os seres humanos em seu cotidiano ou para seus negócios. Uma questão endereçada neste artigo é como dados coletados por meio de dispositivos compatíveis com um padrão aberto de IoT podem ser transformados em informações úteis.

O cenário considerado neste artigo está voltado à educação superior: um laboratório remoto construído com dispositivos IoT. O conceito de laboratório remoto obteve destaque durante a pandemia

de COVID-19, devido à adoção dos protocolos de distanciamento social, inviabilizando o uso presencial da infraestrutura experimental de laboratório nas universidades e faculdades. Exemplos de uso de laboratórios remotos podem ser encontrados na literatura [3, 5, 11, 12].

Um laboratório remoto é constituído por montagens experimentais, que são implantadas no laboratório da instituição de ensino, e *softwares* para acesso remoto a esta montagem. Uma maneira simples de se implementar um laboratório remoto consiste em conectar *desktops* localizados no laboratório com dispositivos IoT, além de instalar algum software de acesso remoto, como o Anydesk [6]. Desta forma, o aluno pode programar, compilar e carregar programas na placa didática do laboratório. Além disso, o uso de *webcams* permite ao aluno ter um retorno visual das mudanças na montagem experimental consequentes de comando feitos por ele. A Figura 1 ilustra esta arquitetura de laboratório remoto.

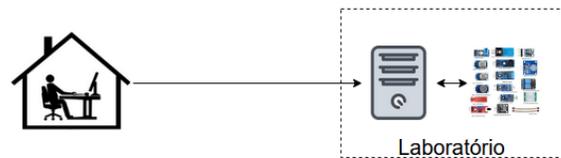


Figura 1: Exemplo de laboratório remoto

Por mais simples que seja a descrição apresentada, ela possui um problema: como garantir a integridade dos dispositivos utilizados? Deixar a montagem experimental ligada 24/7 interfere na vida útil dos dispositivos. Por outro lado, constantes viagens da equipe para o laboratório com o propósito de ligar/desligar a montagem experimental desrespeitam os protocolos de distanciamento social.

Este artigo apresenta um *dashboard open source* com interface *web* para controle de dispositivos IoT baseado no protocolo MQTT (*Message Queuing Telemetry Transport*) [14]. A ferramenta permite ao usuário monitorar e controlar remotamente dispositivos da montagem experimental sem a necessidade de presença física no laboratório.

## 2 TRABALHOS RELACIONADOS

Um resumo das ferramentas disponíveis para criação de *dashboards* IoT é apresentado na Tabela 1, que considera os seguintes aspectos:

- Programação visual: É possível criar fluxos de consumo e processamento de dados por meio de uma interface gráfica.

Ferramenta	Programação Visual	Interface	Suporta MQTT	Grau de Personalização	Código Aberto
Grafana [10]	não	web	não	moderado	sim
MQTT Dash [13]	sim	mobile	sim	baixo	sim
Blynk [7]	sim	mobile	não	baixo	não
Freeboard [8]	não	web	não	moderado	sim
Thingsboard [2]	sim	web	sim*	moderado*	sim**
Solução Proposta	não	web	sim	alto	sim

\* Somente na *Professional Edition*

\*\* Somente na *Community Edition*

**Tabela 1: Resumo das ferramentas de *dashboard***

- Interface: *Web* para aplicações acessíveis pelo *browser*, *Mobile* para aplicações nativas de celulares, encontradas nas lojas do Android ou Apple.
- Suporte MQTT: Suporta o protocolo MQTT nativamente, sem necessitar extensões ou processamento prévio por outra aplicação.
- Grau de personalização: três níveis conforme descrito a seguir.
  - Baixo: Não é possível estender a aplicação. Uso de módulos “caixa preta”, sem possibilidade de modificar a arquitetura conforme a necessidade.
  - Moderado: É possível estender a aplicação com plugins e/ou código arbitrário.
  - Alto: Controle completo da aplicação, sendo possível modificar e adicionar funcionalidades de forma livre. Arquitetura aberta, que pode evoluir e ser ajustada conforme a necessidade.
- Código aberto: O código fonte da aplicação pode ser encontrado em repositórios disponíveis ao público, usualmente sob licenças *Creative Commons*.

O Grafana [10] é uma ferramenta gratuita e *open source* para criação de visualizações interativas de métricas e monitoramento de aplicações. Seu foco primário é a criação de gráficos de séries temporais, sendo compatível com bancos de dados de séries temporais (TSDB) como InfluxDB e TimescaleDB, mas também é possível integrá-lo com quaisquer outras fontes de dados já que a ferramenta é agnóstica à sua origem. A popularidade do Grafana é enorme: seu repositório está entre os 130 mais populares do planeta [9], o que indica a capacidade e confiabilidade dessa solução. Uma carência da ferramenta é relacionada a criação de gráficos que não são relacionados à uma escala de tempo, sendo ferramentas como PowerBI ou Tableau mais propícias para esse tipo de caso de uso, mas *plugins* e extensões podem trazer as funcionalidades extras desejadas.

O MQTT Dash (*MQTT Dashboard*) [13] é um aplicativo desenvolvido para dispositivos Android que permite implementar *dashboards* para dispositivos conectados por meio do protocolo MQTT. Seu uso é gratuito, não havendo um sistema de transações ou de assinatura no aplicativo. Além disso, ele é disponibilizado sob uma licença *Creative Commons*. Inicialmente o usuário precisa selecionar os parâmetros relacionados a um *broker* que ele deseja se conectar, e em seguida selecionar os tópicos que ele irá monitorar pelo *dashboard*. Um destaque é que o MQTT Dash também permite a publicação em determinados tópicos para o controle de atuadores em dispositivos IoT, por exemplo.

O Blynk [7] é uma plataforma IoT criada com o propósito de simplificar projetos desta área. Seu aplicativo pode ser baixado tanto em dispositivos Android, na Google Play quanto em dispositivos iOS na App Store. Possui diversos *widgets* à disposição, e com eles o usuário pode personalizar seu projeto mediante o uso dos chamados “pontos de energia”, possuindo um valor finito para contas gratuitas e ilimitado para contas vinculadas a uma assinatura mensal. Possui uma *Application Programming Interface* (API) para facilitar a integração com aplicações externas por meio de requisições HTTP para mudar o estado de um *widget*, mandar um email ou uma notificação *push* para o dispositivo celular, ou apenas consultar o estado de um *widget*.

O Freeboard [8] é uma ferramenta simples e intuitiva para construção de *dashboards* construído com HTML5, javascript e CSS. É possível executar a aplicação em qualquer servidor *web* como um conjunto de arquivos estáticos. A arquitetura do Freeboard é extensiva, permitindo que usuários criem *widgets* (responsáveis pela visualização dos dados), e *datasources* (fontes de dados) usando fragmentos de javascript arbitrários. Como a aplicação é minimalista, a barreira de entrada é menor quando comparada com seus competidores.

O Thingsboard [2] é uma plataforma completa para IoT, desde a coleta dos dados, processamento customizado usando módulos de programação visual, visualização de gráficos e gerenciamento de dispositivos IoT. A arquitetura do projeto é modular, permitindo usuários substituírem protocolos da camada de transporte, bancos de dados, filas de mensagens, e lógica de processamento de dados. O produto é oferecido em 3 planos diferentes, mas apenas a versão *Enterprise* possui integrações diretas com o protocolo MQTT e a possibilidade de integrações customizadas.

Por fim, a solução proposta é uma interface que suporta MQTT, e possui um alto grau de personalização com alta flexibilidade para expansões futuras, e que é disponibilizada como um recurso de código aberto para uso e expansão pela comunidade de desenvolvedores, apesar de não apresentar a facilidade de programação visual de outras soluções.

### 3 APRESENTAÇÃO DA FERRAMENTA

A Figura 2 apresenta a arquitetura MQTT desenvolvida para um laboratório remoto de eletrônica digital. Este laboratório baseia-se principalmente no uso de placas do tipo *Field-programmable gate array* (FPGA), capazes de simular o comportamento de circuitos lógicos por meio de linguagens de descrição de *hardware*. Por mais que aqui seja dado um enfoque neste cenário de uso, a ferramenta

desenvolvida é flexível o bastante para poder ser utilizada em casos diversos, não se limitando a este.

Para que os tópicos sejam únicos, optou-se por seguir a convenção conforme hierarquia <modelo>/<chipID>/<tópico>, onde o modelo está relacionado à versão do protótipo do dispositivo ou *dashboard* IoT, o chipID é um identificador único, e o tópico é o tipo do tópico considerado.

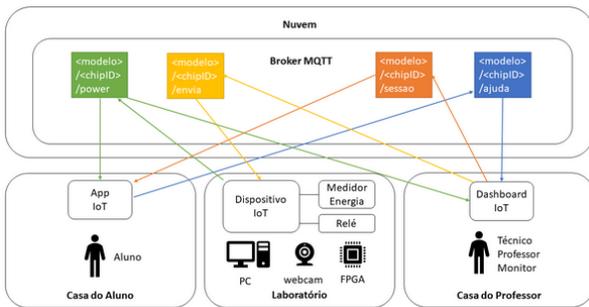


Figura 2: Arquitetura MQTT

Durante uma sessão de uso de um aluno em uma bancada remota, os dados de consumo de energia da placa FPGA são publicados pelo dispositivo IoT, e tanto o professor quanto o aluno podem acompanhar este consumo. O *dashboard* pode controlar o estado de placa FPGA (i.e., ligado ou desligado) publicando no tópico <modelo>/<chipID>/envia. O dispositivo IoT está subscrito neste tópico, e atualiza o estado de seu atuador sempre que recebe uma nova mensagem neste tópico.

O professor pode observar um *dashboard* com informações de todas as bancadas disponíveis. O *dashboard* com dois módulos apresentado na Figura 3 apresenta a situação da bancada (i.e., ocupada ou livre), o tempo restante da sessão atual do aluno, o consumo de energia atual, se o aluno está pedindo ajuda, além de informações para o acesso remoto usando o programa Anydesk e o código de videoconferência da bancada.



Figura 3: Protótipo de *dashboard web* para o professor

Foi necessária a subscrição do *dashboard* IoT em alguns tópicos de interesse, o gerenciamento de conexão, e a especificação do

comportamento após o recebimento de determinadas mensagens nos tópicos em que o *dashboard* está subscrito. Este *dashboard* se comunica diretamente com o *broker* MQTT por meio de uma comunicação *websockets*. Para o teste da ferramenta, dois dispositivos IoT desenvolvidos com a placa ESP8266 foram instalados no Laboratório de Eletrônica Digital da Escola Politécnica da USP com o auxílio de um técnico de laboratório.

Além do *dashboard web*, uma interface *mobile* para uso por professores, técnicos e monitores foi construída usando o MQTT Dash por sua compatibilidade com o protocolo MQTT, e sua facilidade de aprendizado. A Figura 4a apresenta a configuração de um tópico de atuador (em que o *mobile* deve publicar), enquanto a Figura 4b apresenta uma configuração de um tópico de sensor (em que o *mobile* deve estar subscrito). Por fim, a Figura 4c apresenta o resultado com cinco possíveis bancadas para controle do estado da FPGA e monitoramento do consumo de energia. O aluno pode pedir ajuda ao professor por meio de outra interface adicional, o App IoT, que permite observar o tempo restante na sua sessão e o consumo atual da placa FPGA da bancada remota.

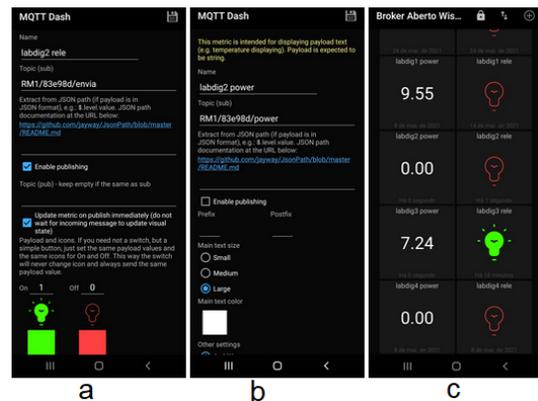


Figura 4: Protótipo MQTT Dash [13] para o professor

O professor monitora o consumo das bancadas e pode ligar e desligar a placa FPGA. Ele também pode acessar remotamente o computador da bancada remota por meio do Anydesk e visualizar os componentes com a *webcam*. Na Figura 5a, o professor observa que a placa da FPGA da bancada está desligada, com um consumo de 0,00 Watts. Após o professor ligar a placa FPGA por meio de seu aplicativo, ele observa o consumo de 7,01 Watts da placa FPGA que acabou de ligar, conforme Figura 5b.

A Figura 6a mostra a situação antes do aluno pedir ajuda. Neste caso, o professor pode observar que a bancada está disponível, que a placa FPGA está desligada (i.e., pela *webcam* e pelo *dashboard web* com o consumo de 0.0W). A situação após o aluno pedir ajuda ao professor é apresentada pela Figura 6b. O aluno pressionou o botão de ajuda, e o Professor recebe este pedido no seu *dashboard*.

Após o professor iniciar a sessão de uso por meio de seu *dashboard*, a placa FPGA é ligada e o contador de tempo da sessão é iniciado, conforme Figura 6c. O aluno recebe as informações de consumo da FPGA e de tempo restante na sessão no seu aplicativo, e pode desativar o botão de ajuda e prosseguir com suas atividades práticas usando o acesso remoto e a videoconferência da bancada

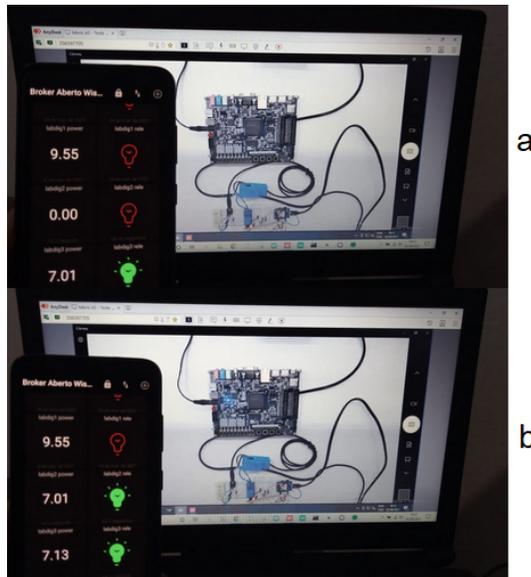


Figura 5: Teste de integração com dispositivo IoT e *dashboard mobile* do professor

para se comunicar com sua equipe ou com o professor. Ao finalizar a contagem, a FPGA é desligada automaticamente.

O *dashboard web* para o professor está disponível sob uma licença *Creative Commons* no [GitHub](#). Um vídeo de demonstração da ferramenta está disponível no [Youtube](#).

#### 4 CONCLUSÃO E TRABALHOS FUTUROS

Este artigo apresentou um *dashboard* IoT para o controle e monitoramento de montagens experimentais em um laboratório remoto, com o uso do protocolo aberto MQTT. Também apresentou um comparativo entre a solução disponibilizada como código aberto e diferentes ferramentas para criação de *dashboards*. É esperado que a ferramenta auxilie na concepção de novas interfaces para o ensino remoto, ou como um exemplo de uma implementação de um *dashboard* compatível com o padrão aberto MQTT, desenvolvido em HTML/javascript. Para a utilização em outras disciplinas, os tópicos de monitoramento e controle devem ser adaptados, porém o uso do protocolo MQTT e suas interfaces compatíveis se mantêm para outras disciplinas e experimentos.

Como trabalhos futuros, há oportunidades de explorar novas propostas de valor relacionadas à aspectos pedagógicos para o estudo de caso apresentado, além de complementar a etapa de validação com testes de usabilidade e novas iterações para fomentar a evolução da solução IoT, a partir das percepções sobre o uso da ferramenta por técnicos, professores e alunos. Adicionar aspectos de segurança como autenticação e controle de acesso também são oportunidades.

#### REFERÊNCIAS

- [1] Kevin Ashton et al. 2009. That 'internet of things' thing. *RFID journal* 22, 7 (2009), 97–114.
- [2] The ThingsBoard Authors. 2021. Thingboard. <https://thingsboard.io/>. Acesso: 29/05/2021.

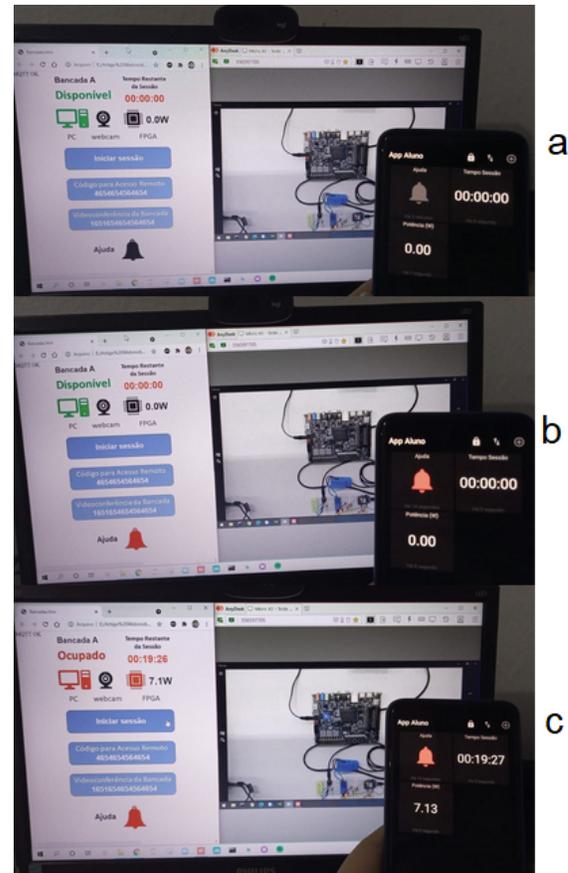


Figura 6: Teste de integração com dispositivo IoT, interface *mobile* do aluno e *dashboard web* do professor

- [3] Marcos Henrique de Morais Golinelli, Juarez Bento da Silva, and Olga Yevseyeva. 2018. ARQUITETURA DE DISPOSITIVOS INTELIGENTES PARA LABORATÓRIOS REMOTOS: UMA REVISÃO SISTEMÁTICA DA LITERATURA. *RETEC-Revista de Tecnologias* 11, 2 (2018).
- [4] Ericsson. 2021. IoT connections outlook. <https://www.ericsson.com/en/mobility-report/dataforecasts/iot-connections-outlook>. Acesso: 29/09/2021.
- [5] Javier Garcia-Zubia, Jordi Cuadros, Susana Romero, Unai Hernandez-Jayo, Pablo Orduna, Mariluz Guenaga, Lucinio Gonzalez-Sabate, and Ingvar Gustavsson. 2016. Empirical analysis of the use of the VISIR remote lab in teaching analog electronics. *IEEE Transactions on Education* 60, 2 (2016), 149–156.
- [6] AnyDesk Software GmbH. 2021. AnyDesk. <https://anydesk.com/pt>. Acesso: 29/05/2021.
- [7] Blynk Inc. 2021. Blynk. <https://blynk.io/>. Acesso: 29/05/2021.
- [8] Bug Lab Inc. 2021. Freeboard. <https://freeboard.io/>. Acesso: 29/05/2021.
- [9] Takashi Kokubun. 2021. Repositories Ranking. <https://gitstar-ranking.com/repositories>. Acesso: 02/06/2021.
- [10] Grafana Labs. 2021. Grafana. <https://grafana.com/>. Acesso: 29/05/2021.
- [11] Imad Mougharbel, Ali El Hajj, HASSAN Artail, and Chadi Rimam. 2006. Remote lab experiments models: A comparative study. *International Journal of Engineering Education* 22, 4 (2006), 849.
- [12] Fretz Sievers Jr, José Silvério E Germano, and Felipe de Almeida. 2007. WEBLAB—Um laboratório remoto para experimentos de Física. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, Vol. 1. 267–270.
- [13] Routix Software. 2021. MQTT Dash. <https://play.google.com/store/apps/details?id=net.routix.mqttdash>. Acesso: 29/09/2021.
- [14] OASIS Standard. 2014. MQTT version 3.1. 1. URL <http://docs.oasis-open.org/mqtt/mqtt/v3.1> (2014).