

# OTALab: um ambiente de experimentação remota de protocolos e aplicações em Internet das Coisas\*

Enzo B. Cussuol<sup>1</sup>, Lucas L. S. Sachetti<sup>1</sup>, Bruno P. Santos<sup>2</sup>, Vinicius F. S. Mota<sup>1</sup>

<sup>1</sup>Departamento de Informática  
Universidade Federal do Espírito Santo Vitória (UFES) – Brasil

<sup>2</sup>Departamento de Computação e Sistemas  
Universidade Federal de Ouro Preto (UFOP) – MG – Brasil.

{enzo.cussuol, lucas.sachetti}@edu.ufes.br

bruno.ps@ufop.edu.br, vinicius.mota@inf.ufes.br

**Abstract.** *This paper presents the OTALab, a tool for creating and deploying experimentation environments of Internet of Things (IoT) applications in low-cost microcontrollers. The OTALab aims for fast deployment and configuration of an experimentation environment testbed. OTALab has two users profiles: administrators and experimenters. For the former, the OTALab exposes an admin interface able to add/remove IoT devices, services, and functionalities to the system. For the latter, the experimenters, can visualize the available services in each device and submit their own code to the devices through a command line or a Web application. OTALab receives the source code, compiles it to the specific microcontroller, and updates the device's firmware through the Over the Air (OTA) paradigm. The OTALab is composed by a library of microcontrollers, a device management server, and a Web application. The tool can be instantiated locally or in a distributed fashion, improving its flexibility.*

**Resumo.** *Este artigo apresenta o OTALab, uma ferramenta para criação e implantação de ambientes de experimentação de aplicações de Internet of Things (IoT) em microcontroladores de baixo custo. O OTALab visa a rápida implantação e configuração de um testbed de ambiente de experimentação. O OTALab possui dois perfis de usuários: administradores e experimentadores. Para os primeiros, o OTALab expõe uma interface de administração capaz de adicionar/remover dispositivos IoT, serviços e funcionalidades ao sistema. Já os experimentadores podem visualizar os serviços disponíveis em cada dispositivo e enviar seu próprio código aos dispositivos via linha de comando ou interface Web. O OTALab recebe o código-fonte, compila-o para o microcontrolador específico e atualiza o firmware do dispositivo através do paradigma Over the Air (OTA). O OTALab é composto por uma biblioteca de microcontroladores, um servidor de gerenciamento de dispositivos e uma aplicação Web. A ferramenta pode ser instanciada localmente ou de forma distribuída, melhorando sua flexibilidade.*

---

\*O presente trabalho foi realizado com apoio financeiro da FAPES, FAPEMIG, FAPESP - Porvir-5G (Grant #2020/05182-3), CNPq e Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil(CAPES) - Código 001

## 1. Introdução

A Internet das Coisas, do inglês *Internet of Things (IoT)* é um termo amplo utilizado para caracterizar uma rede formada por objetos com identificação única e integração transparente à Internet [Atzori et al. 2010]. De forma genérica, os autores em [Atzori et al. 2017] definem IoT como um arcabouço conceitual que utiliza dispositivos heterogêneos e interconectados para prover compartilhamento de informação em escala global.

As atividades de pesquisa e ensino em IoT comumente exigem acesso físico aos dispositivos. Entretanto, a configuração, instalação ou manejo destes dispositivos podem exigir um conhecimento amplo, que vai da micro-eletrônica, montagem de microcontroladores e sensores aos protocolos e aplicações IoT. Adicionalmente, ensino e pesquisas que utilizam dispositivos físicos foram comprometidos nas instituições que adotaram trabalho remoto durante a pandemia COVID-19. Uma forma de mitigar tais problemas é por meio da utilização de ambientes de experimentação e simulação, conhecido com *testbeds*, os quais são projetados para avaliar novas tecnologias, protocolos, aplicações e para fins educativos [Cosmi and Mota 2019].

*Testbeds* se caracterizam por serem um ambiente de experimentação e simulação nos quais usuários podem testar suas aplicações. Tais ambientes de experimentação já são utilizados para os estudos de sistemas distribuídos, como o PlanetLab [Peterson et al. 2003] e integração de camadas físicas, como o RNP/EU FUTEBOL [Both et al. 2019]. No contexto de IoT, o uso de *testbeds* torna-se essencial para desenvolvedores e pesquisadores analisarem as mais novas tecnologias, algoritmos, técnicas e protocolos [Chernyshev et al. 2017]. Entre as principais propostas de *testbed* para a disponibilização de dispositivos heterogêneos, destacam-se o MoteLab [Werner-Allen et al. 2005], INDRYA 2 [Appavoo et al. 2018], FIT IoT-LAB [Adjih et al. 2015]. Contudo, apesar de serem robustos, os *testbeds* possuem limitações de disponibilidade, alta curva de aprendizado, e, muitas vezes, falta flexibilidade quando os experimentos exigem privilégios administrativos de uso. Por outro lado, devido aos baixos custos de dispositivos com microcontroladores e redes embarcados, é cada dia mais comum que grupos de pesquisa tenham um conjunto destes dispositivos disponíveis. Diferentemente do uso dos *testbeds* citados acima, ao possuir os dispositivos, o pesquisador tem mais controle sobre o gerenciamento destes dispositivos, tais como controle de acesso, configurações, sensores, serviços, topologia da rede, além de depuração local. Este trabalho se encaixa nestas lacunas.

Este artigo propõe o OTALab, uma ferramenta para criação e implantação de *testbed* para o estudo de protocolos e aplicações IoT em dispositivos com microcontroladores. O OTALab permite criar um *testbed* de forma rápida e com pouca configuração, possuindo dois perfis de usuários totalmente configuráveis: *Administradores* e *Experimentadores*. Os *Administradores* possuem acesso a uma interface de uso capaz de adicionar ou remover dispositivos IoT, serviços e funcionalidades ao sistema. Já usuários *Experimentadores* acessar os serviços de cada dispositivo bem como enviar códigos para os dispositivos. A interação com o sistema pode ser feita via comandos usando *OTALab Command Line Interface (OTALab-CLI)* ou via interface web intuitiva *OTALab Graphical User Interface (OTALab-GUI)*. O OTALab implementa em sua biblioteca o paradigma Over the Air (OTA) [Arakadakis et al. 2021], no qual os dispositivos recebem atualizações de *firmware* por meio de uma comunicação sem fio. Deste modo, o *testbed* pode definir livremente

a localização dos dispositivos no ambiente, bem como a topologia da rede. A principal diferença e proposta deste trabalho em relação aos *testbeds* supracitados é permitir que pesquisadores montem seu próprio *testbed*. Para tanto, OTALab permite controle sobre disponibilidade, depuração e ajuste fino administrativo do sistema, enquanto mantém baixa a curva de aprendizagem. Desta forma, pesquisas relacionadas a algoritmos, protocolos e aplicações em IoT podem ser realizadas in loco ou remotamente.

O OTALab está sendo utilizado como ferramenta de ensino nas disciplinas relacionadas à IoT da Universidade Federal do Espírito Santo e do Programa de Pós Graduação em Informática/UFES. Além, é claro, do foco de pesquisa, para análise e desenvolvimento de protocolos e aplicações IoT.

O restante deste artigo está organizado como se segue: A Seção 2 detalha a arquitetura do OTALab. A Seção 3 apresenta os *links* para o código fonte e vídeo de instalação e uso. As discussões finais, bem como os trabalhos futuros, são apresentadas na Seção 4.

## 2. O OTALab

O OTALab proporciona aos pesquisadores um ambiente de experimentação e configuração rápida. Além disso, atualização OTA dos *firmwares* dos dispositivos IoT em rede. O OTALab possui dois perfis na utilização: i) O *Administrador*, que pode adicionar e remover dispositivos, bem como adicionar e remover serviços/funcionalidades oferecidos por cada dispositivo. Desta forma, cada dispositivo pode ter um conjunto de sensores e/ou rádios distintos. ii) O *Experimentador*, que é capaz de visualizar os dispositivos ativos e seus respectivos serviços e funcionalidades. O experimentador pode submeter códigos para qualquer um dos dispositivos IoT disponíveis, sendo exigido apenas que se utilize uma biblioteca provida pelo OTALab, a qual é responsável por manter a conectividade e definir parâmetros da automatização dos processos de compilação e atualizações de *firmware*. O Administrador e o experimentador podem interagir com o *testbed* via linha de comando através do OTALab-CLI ou via interface web OTALab-GUI.

### 2.1. Visão geral da arquitetura proposta

A arquitetura do OTALab e a interação entre componentes são apresentados na Figura 1. A interface Web viewer OTALab-GUI fornece aos usuários do sistema um modo rápido, intuitivo, remoto e de baixa curva de aprendizagem aos principais recursos da ferramenta. Esses recursos são implementados majoritariamente pelo módulo OTALab Core. Esse é o módulo responsável pelo repositório de código e configurações do sistema, configuração dos dispositivos, descobertas de dispositivos e serviços, bem como é o gestor do processo de envio de códigos para os dispositivos. Para que um dispositivo possa ser descoberto e ter seu *firmware* atualizado via OTA, ele precisa estar executando um código padrão, o qual deve utilizar os recursos da biblioteca OTALab Lib.

O Web viewer OTALab-GUI expõe visões distintas para o Administrador e o Experimentador do sistema. De modo geral, o Administrador define as configurações de ambiente, tais como SSID e senha da rede Wi-Fi em que estarão os dispositivos, bem como configura um novo dispositivo, definindo o modelo do dispositivo e sensores disponíveis. Na primeira configuração, o dispositivo precisa estar conectado via USB ao OTALab Core para que seja feita a compilação e instalação de um código padrão no dispositivo, a qual contém a OTALab Lib. Após, os dispositivos podem ser descobertos e

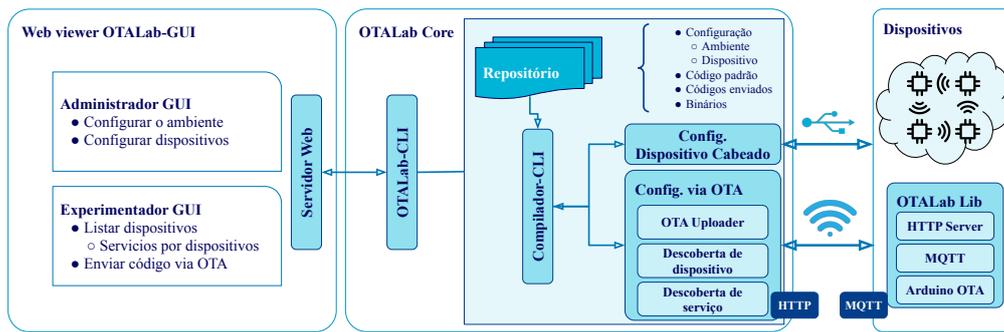


Figura 1. Visão geral da arquitetura do OTALab.

atualizados via OTA. Já a visão do Experimentador apresenta a lista de dispositivos e todos os serviços disponíveis ou que podem ser implementados. Por exemplo, um dispositivo contendo um sensor de temperatura poderia anunciar o modelo do sensor ou o nome do serviço que retorna a temperatura aferida via HTTP. Ressalta-se que um experimentador mais experiente poderia interagir com o *testbed* via comandos através do OTALab-CLI.

## 2.2. OTALab Lib: Biblioteca executada nos Dispositivos

A OTALab Lib é uma biblioteca que deve ser adicionada e instanciada no dispositivo para que seus serviços possam ser descobertos, além de possibilitar a atualização de código OTA. A biblioteca foi desenvolvida para uso com Arduino IDE com o intuito de abstrair do desenvolvedor experimentador as questões de configurações de rede sem fio, uso do HTTP e MQTT para comunicação remota com os dispositivos. As informações dessas configurações são definidas pelo Administrador. O objetivo da OTALib é garantir que o dispositivo crie um servidor HTTP, o qual permite a consulta de informações, por exemplo, a lista de serviços. Já o MQTT é utilizado para descobrir de forma rápida quais dispositivos estão na rede. Ao iniciar, o dispositivo registra-se em um tópico MQTT, ao receber uma mensagem neste tópico do OTALab Core, o dispositivo responde com seu nome e endereço IP.

Vale ressaltar que o experimentador necessita inserir apenas algumas linhas em seu código para utilizar um dispositivo no *testbed*. O trecho de código abaixo, escrito em C++, ilustra a inclusão da biblioteca (linha 1), a instanciação de um dispositivo (linha 3), e a inicialização dos serviços (linha 6-10). Esses serviços iniciam a conexão com a rede sem fio do *testbed*, um microserviço HTTP e uma conexão MQTT com um *broker*.

```

1 #include <OTANetworkDevice.h>
2
3 OTANetworkDevice* device = new OTANetworkDevice(esp8266D1Mini);
4
5 void setup() {
6     device->setWiFiNetworkName(WIFI_NETWORK_NAME);
7     device->setWiFiNetworkPassword(WIFI_NETWORK_PASSWORD);
8     device->setBrokerIP(BROKER_IP);
9     device->setName(DEVICE_NAME);
10    device->setup();
11 }
12 void loop() {
13     device->handle();
14 }

```

Para este trabalho, a OTALab Lib é compatível com os microcontroladores ESP8266 e ESP32. A OTALab Lib utiliza outras três bibliotecas: WI-FI<sup>1</sup>, em dispositivos ESP32, ou ESP8266WiFi<sup>2</sup>, em dispositivos com ESP8266, para funções de conexão via WiFi; ArduinoOTA<sup>3</sup>, responsável pela atualização Over the Air; PubSubClient<sup>4</sup> que implementa o protocolo MQTT, o qual permite que dispositivos se conectem à um *broker* e assim é possível se inscrever e publicar em tópicos. É importante destacar que o *broker* pode estar na rede local ou na Internet/Nuvem.

### 2.3. OTALab Core

O OTALab Core é responsável pelo gerenciamento e orquestração de todos os dispositivos que compõem o *testbed*. O OTALab Core é composto por um repositório para armazenar configurações e códigos enviados, compiladores específicos para os microcontroladores suportados, suporte para envio de *firmware* via cabo, além da descoberta de dispositivos, seus respectivos serviço e pela atualização de *firmware* via OTA. A seguir detalhamos o funcionamento do OTALab Core quanto ao papel do administrador e do experimentador.

#### 2.3.1. Configuração do OTALab e adição de dispositivos

O Administrador do OTALab é responsável por configurar cada dispositivo que irá compor o *testbed*. Isso é realizado conectando-se, inicialmente, de modo físico um dispositivo ao OTALab Core. Neste processo, um código padrão de configuração será instalado no dispositivo para que seja possível reconhecê-lo e gerenciá-lo remotamente via rede. Para tanto, exige-se apenas que o Administrador defina o modelo do dispositivo, a existência de sensores acoplados e em quais saídas (pinos) dos dispositivos estão conectados, o endereço do *broker* MQTT, bem como o SSID e senha da rede sem fio que o dispositivo deve se conectar.

A Figura 2 sumariza o papel do administrador no OTALab. A Figura 2a apresenta o diagrama de sequência da configuração de um dispositivo. Após enviar as configurações descritas acima, o OTALab Core armazena as informações do dispositivo, altera as variáveis de ambiente no arquivo de cabeçalho da OTALib e compila para o dispositivo específico. Em seguida, envia-se o código padrão para o dispositivo via porta USB. A Figura 2b apresenta a visão do Administrador, através da OTALab-GUI, para configurar um dispositivo.

#### 2.3.2. Descoberta, Seleção de dispositivos e envio de código

Como um Experimentador é possível visualizar os dispositivos ativos no momento. Para cada dispositivo, apresenta-se informações tais como o seu nome, modelo, IP, lista de serviços e qual código está sendo executado. O Experimentador também pode selecionar um ou mais desses dispositivos para realizar atualização de código fonte, via OTA ou

---

<sup>1</sup><https://github.com/espressif/arduino-esp32/tree/master/libraries/WiFi>

<sup>2</sup><https://github.com/esp8266/Arduino/tree/master/libraries/ESP8266WiFi>

<sup>3</sup><https://github.com/jandrassy/ArduinoOTA>

<sup>4</sup><https://github.com/knolleary/pubsubclient>

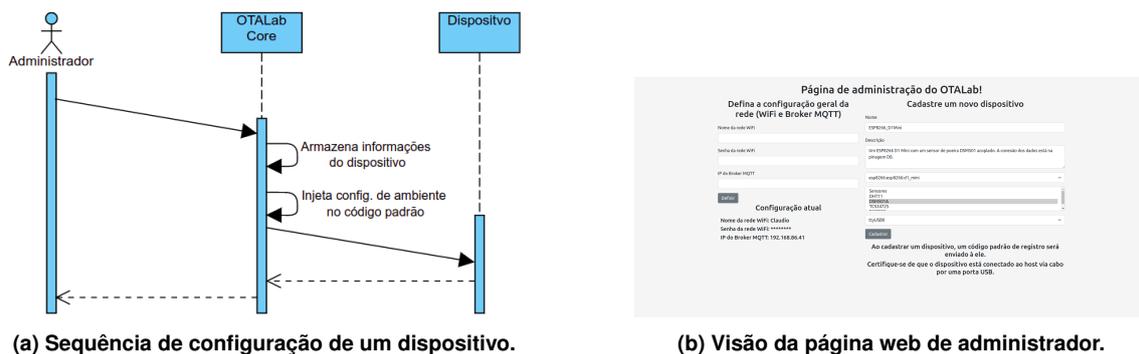


Figura 2. Configuração de um dispositivo no OTALab.

cabo. A Figura 3 apresenta o diagrama de sequência para a descoberta de dispositivos e serviços oferecidos. Essas funcionalidades são realizadas pelos seguintes módulos:

**Descoberta de dispositivo:** Consiste em descobrir os endereços IPs dos dispositivos conectados ao OTALab. Uma forma de fazer a descoberta de dispositivos é por meio de varredura de endereços dentro de uma mesma rede. No entanto, isso obrigaria que o OTALab e os dispositivos estivessem em uma mesma rede sem fio. Para contornar este problema, foi utilizado o protocolo MQTT. Neste caso, os dispositivos e o OTALab Core se inscrevem em tópico de requisição e resposta em um mesmo *broker*. Quando o módulo é executado publica-se uma mensagem em um tópico de requisição, nos quais todos os dispositivos são inscritos. Os dispositivos, ao receberem a mensagem, publicam uma resposta contendo seus endereços IP, os quais são então armazenados no servidor. Vale destacar que o *broker* MQTT é o responsável intermediação entre o OTALab Core e os dispositivos. Neste trabalho, foi utilizado o Mosquitto como um *broker* local, executando em conjunto com o OTALab Core. Contudo, o endereço do *broker* é configurável.

**Descoberta de serviços:** Esse módulo realiza requisições HTTP nos endereços IP armazenados pelo módulo de descoberta de dispositivos. As requisições visam descobrir quais serviços são ofertados por um dispositivo. Por exemplo, um dispositivo pode retornar o conjunto de sensores disponíveis (modelos e a quais portas lógicas estão conectados). Dessa forma, o experimentador tem a visão do tipo de aplicação que ele pode implementar com base nos recursos dos dispositivos. O experimentador também pode implementar e disponibilizar serviços em um dispositivo.

A OTALab disponibiliza um serviço de *Ping* para que seja possível a alcançabilidade dos dispositivos instalados. Com isto também é possível gerenciar e visualizar os dispositivos que estão ativos e disponíveis na rede.

**OTAUploader:** Realiza os procedimentos necessários para enviar o código do usuário para o dispositivo. O código transferido ao OTALab Core é armazenado em um repositório e automaticamente compilado para o tipo de dispositivo especificado. Neste processo, a biblioteca OTALab Lib também é incluída. Adicionalmente, é possível que o código também seja compilado para um dispositivo teste (conectado via porta USB) que atua como um “*Sandbox*”, isto é, verifica que o código enviado contém a biblioteca da OTALab Lib e retorna as requisições de serviço. Embora esta etapa seja opcional e configurado pelo Administrador, isso garante que os dispositivos fiquem inacessíveis via OTA devido a erros do experimentador. Após resposta positiva do processo de compilação pelo

dispositivo teste, o *firmware* é enviado, via OTA, ao dispositivo escolhido. A Figura 4a apresenta o diagrama de sequência para envio de *firmware* aos dispositivos selecionados.

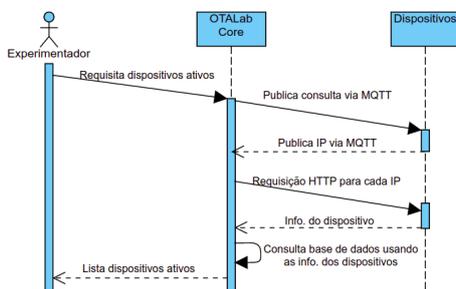
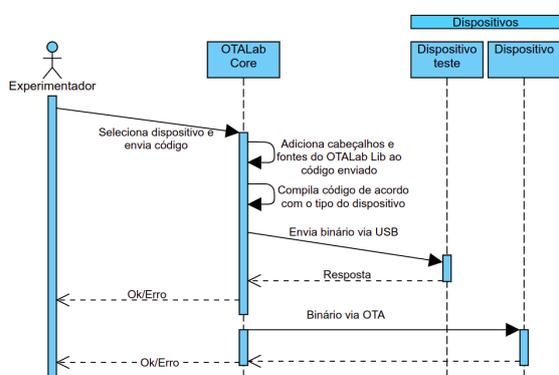


Figura 3. Diagrama de sequência da descoberta de dispositivo e de serviços.



(a) Diagrama de sequência da atualização de *firmware*.



(b) Visão da página web do Experimentador.

Figura 4. Visão do Experimentador

Note que o experimentador pode realizar a submissão de código via OTALab-GUI ou OTALab-CLI. A interface Web, apresentada na Figura 4b, abstrai todos os detalhes de compilação e especificidade de cada dispositivo. Permitindo que o experimentador foque apenas no código do experimento. Um caso de uso considerando a visão dos experimentadores é apresentado por meio de um *walk-through* no git do projeto.

### 3. Materiais Online e forma de apresentação

**Código fonte:** <https://github.com/enzocussuol/OTALab>

**Vídeo com instruções para instalação:** <https://youtu.be/sTEk3hqMuC0>

#### 3.1. Forma de Apresentação

A apresentação da ferramenta será realizada de forma remota com a demonstração de um caso de uso. A demonstração apresentará a configuração de um *testbed* com pelo menos quatro microcontroladores ESP8266 e ESP32 contendo sensores variados, tais como temperatura, pressão materiais particulados, entre outros.

O OTALab será utilizado para envio de códigos de experimentos pré-configurados aos nós instalados na etapa apresentada anteriormente.

## 4. Conclusão

Neste artigo, foi apresentado uma ferramenta que visa facilitar o gerenciamento de dispositivos IoT, compostos por microcontroladores, de modo remoto. O OTALab se destaca por seu potencial de aplicação, que, como discutido, é muito amplo, pela sua praticidade e simplicidade. Além disso, o OTALab cumpre o seu objetivo principal que é o de servir como uma ferramenta de ensino remoto para disciplinas relacionadas à IoT.

Como trabalhos futuros, pretende-se adicionar camadas de segurança e prevenção contra códigos mal formatados que podem eventualmente ser enviados para os dispositivos. Ademais, os autores pretendem adicionar suporte a outras tecnologias de comunicação, tais como LoraWan e Bluetooth Low-Energy (BLE). Por fim, almeja-se dar suporte a mais classes de dispositivos.

## Referências

- Adjih, C., Baccelli, E., Fleury, E., Harter, G., Mitton, N., Noel, T., Pissard-Gibollet, R., Saint-Marcel, F., Schreiner, G., Vandaele, J., et al. (2015). FIT IoT-LAB: A large scale open experimental IoT testbed. In *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, pages 459–464. IEEE.
- Appavoo, P., William, E. K., Chan, M. C., and Mohammad, M. (2018). Indriya2: A heterogeneous wireless sensor network (wsn) testbed. In *International Conference on Testbeds and Research Infrastructures*, pages 3–19. Springer.
- Arakadakis, K., Charalampidis, P., Makrogiannakis, A., and Fragkiadakis, A. (2021). Firmware over-the-air programming techniques for iot networks—a survey. *ACM Computing Surveys (CSUR)*, 54(9):1–36.
- Atzori, L., Iera, A., and Morabito, G. (2010). The internet of things: A survey. *Computer networks*, 54(15):2787–2805.
- Atzori, L., Iera, A., and Morabito, G. (2017). Understanding the internet of things: definition, potentials, and societal role of a fast evolving paradigm. *Ad Hoc Networks*, 56:122–140.
- Both, C., Guimaraes, R., Slyne, F., Wickboldt, J., Martinello, M., Dominicini, C., Martins, R., Zhang, Y., Cardoso, D., Villaca, R., et al. (2019). Futebol control framework: Enabling experimentation in convergent optical, wireless, and cloud infrastructures. *IEEE Communications Magazine*, 57(10):56–62.
- Chernyshev, M., Baig, Z., Bello, O., and Zeadally, S. (2017). Internet of things (iot): Research, simulators, and testbeds. *IEEE Internet of Things Journal*, 5(3):1637–1647.
- Cosmi, A. B. and Mota, V. F. (2019). Uma análise dos protocolos de comunicação para internet das coisas. In *Anais do III Workshop de Computação Urbana*, pages 153–166. SBC.
- Peterson, L., Anderson, T., Culler, D., and Roscoe, T. (2003). A blueprint for introducing disruptive technology into the internet. *ACM SIGCOMM Computer Communication Review*, 33(1):59–64.
- Werner-Allen, G., Swieskowski, P., and Welsh, M. (2005). Motelab: A wireless sensor network testbed. In *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.*, pages 483–488. IEEE.