

# Perspectivas para o uso do *Node-Red* no Ensino de IoT

Fernando Ferreira Passe

Departamento de Informática  
Universidade Federal de Viçosa  
Viçosa, MG  
fernando.passe@ufv.br

Vanessa Cristiny Rodrigues Vasconcelos

Departamento de Informática  
Universidade Federal de Viçosa  
Viçosa, MG  
vanessa.vasconcelos@ufv.br

Michael Canesche e Ricardo Ferreira

Departamento de Informática  
Universidade Federal de Viçosa  
Viçosa, MG  
{michael.canesche, ricardo}@ufv.br

**Resumo**—Atualmente, uma possibilidade para o ensino de sistemas embarcados está diretamente ligada ao ensino de internet das coisas ou IoT (*Internet of things*), e este por sua vez exige diversos conhecimentos multidisciplinares. A conexão de dispositivos com a rede ainda exige conhecimento sobre protocolos web e construção de web servers. Este artigo propõe uma metodologia de ensino com exemplos simples e diversificados baseados no uso da ferramenta *Node-RED*. Esta nova ferramenta proposta pela IBM simplifica o projeto de IoT com diversas abstrações e ao mesmo tempo permite explorar tópicos avançados no ensino e pesquisa. O projeto é baseado em fluxos e apesar de recente, a ferramenta já disponibiliza muitos recursos aos programadores. Além do uso de *Node-RED* para ensino, este artigo também sugere o uso de MQTT (*Message Queue Telemetry Transport*) como protocolo base de comunicação.

**Index Terms**—Computação Ubíqua, Computação Pervasiva

## I. INTRODUÇÃO

A Internet das Coisas (IoT) é uma realidade e deve fazer parte da grade curricular dos cursos de Ciência da Computação, Engenharia da Computação e Sistemas de Informação. Conforme previsto por Mark Weiser em seu artigo "*The Computer for the 21st Century*" [1], a computação está se infiltrando em locais que antes eram impossíveis e hoje já faz parte do cotidiano. Os equipamentos comunicam-se localmente ou pela Internet, adicionando um grau de complexidade ao ensino de sistemas embarcados e redes. Em um futuro próximo, todos os equipamentos estarão conectados e farão parte de algum aplicativo. Para os estudantes iniciantes é difícil buscar respostas para vários assuntos devido ao excesso de informações fragmentadas, por exemplo, questões como "por onde começar", "como lidar com projetos de sistemas com vários padrões de conectividade e serviços", etc., assolam a todos.

A área de IoT exige conhecimento multidisciplinar para a construção e conexão de novos dispositivos e para lidar com as novas formas de interagir com o usuário. Plataformas de hardware e software para prototipagem e testes de sistemas como *Arduino* e *Raspberry Pi* são opções viáveis para o ensino, devido ao seu baixo custo e sua usabilidade. Este artigo propõe utilizar o ambiente *Node-RED* [2] desenvolvido pela IBM para o ensino de IoT. A ferramenta *Node-RED* torna o ensino de IoT uma tarefa simples pois permite aos programadores modelar, implementar e validar diversas aplicações

usando uma interface interativa, amigável e com uma curva rápida de aprendizado. A metodologia utilizada é a de projetos e ensino com exemplos.

Este documento está estruturado da seguinte forma: primeiramente, na seção II é apresentada a ferramenta *Node-RED*. Na seção III é introduzido o protocolo MQTT (*Message Queue Telemetry Transport*), um padrão utilizado em projetos com IoT. Posteriormente, na seção IV, os exemplos propostos detalham o uso do *Node-RED* além de mostrarem como desenvolver projetos e como adicionar códigos existentes ao projeto. Finalmente, há sugestão de documentação complementar e comentários finais para o ensino.

## II. NODE-RED

Criar soluções IoT envolve conhecimentos de eletrônica, redes e programação, no qual um dos fatores mais importantes é lidar com as diversas conexões com dispositivos (embarcados, servidores, *smartphones*) através de vários serviços na internet, sendo também necessário o desenvolvimento de programas para enviar, controlar e analisar os dados capturados. Assim para suprir as necessidades desse nicho e projetar os sistemas de IoT surgiram diversas ferramentas [3], [4] e entre elas está o *Node-RED* [2], [5], [6].

O *Node-RED* é um software de código aberto desenvolvido pela IBM que permite a programação através de fluxos (*flow-based programming*), usando uma interface no navegador que pode ser vista na Figura 1.

A plataforma possui vários nós com diferentes funcionalidades que podem ser conectados de forma coerente permitindo a passagem do fluxo de informações e criando aplicações. O princípio é simples, cada nó possui uma funcionalidade bem definida e a maioria dos nós abstrai a implementação do programador. A Figura 2 apresenta o conjunto de nós que são instalados em conjunto com o *Node-RED*. Através do ambiente, o programador projeta diversos fluxos de informação naturalmente assíncronos e paralelos com orientação a eventos. Há diversas possibilidades para a criação destas aplicações com os nós disponíveis, como exemplos de funcionalidades podemos enumerar: conexão com *Twitter*, *Facebook*, *Telegram*, protocolos de rede (udp, tcp, http, mqtt, etc.), conexão local serial, conexão com banco de dados SQL e Não-SQL como *MongoDB*, *mySQL*, etc., painel para

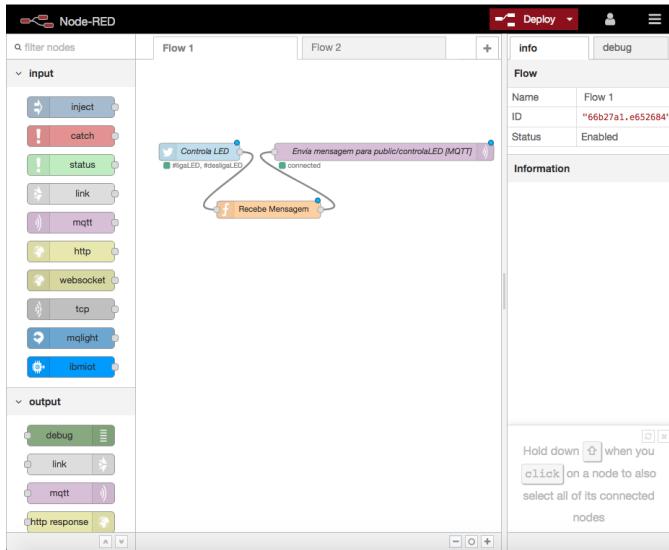


Figura 1. Exemplo da interface do software.

visualização de dados com diversos recursos (gráficos, tabelas, acionadores, etc.), envio de e-mails, conexão com *Chatbots*, processamento de áudio e vídeo, conexão com o sistema de arquivos, análise de dados com recursos simples e complexos, implementação de funções em *Javascript*, *Python* e outras linguagens, funcionalidades específicas para *Raspberry Pi* e *Arduino*, conversão de texto em áudio e vice-versa, ou seja, existe um potencial de utilização para vários recursos com fácil integração e conectividade em uma interface simples, rápida e acessível através de um navegador.



Figura 2. Alguns nós disponíveis no *Node-RED*.

Há diferentes formas para a utilização do *Node-RED*, este

projeto utiliza duas. A primeira é o *Bluemix* [7], plataforma desenvolvida pela IBM para computação na nuvem que possui cota gratuita em seus programas universitários. A interface web para o *Node-RED* executando no *Bluemix* pode ser vista na Figura 1. Para facilitar a busca por informações adicionais, este trabalho concentrou em uma página web [8] uma seleção de links que indicam onde buscar maiores informações sobre o *Bluemix* e sobre outros sistemas que serão apresentados durante o artigo. Outra forma de utilização do *Node-RED* é instalá-lo em um ambiente local ou em um servidor privado, que pode ser um *Raspberry Pi* ou um computador, conforme feito na Universidade Federal de Viçosa [8]. As vantagens desta segunda opção são a gratuidade e o maior controle sobre o serviço, as informações sobre a instalação podem ser obtidas no *GitHub* do projeto [8].

### III. MQTT

Em conjunto com o ambiente *Node-RED*, este artigo ilustra exemplos baseados no protocolo MQTT [9], o qual vem se popularizando pelas suas funcionalidades adequadas ao suporte em IoT. Nos exemplos apresentados, o MQTT será utilizado para troca de mensagens entre os nós do *Node-RED* e os dispositivos embarcados (como *Arduino/ESP8266*, *Raspberry Pi*, entre outros). Lembrando que um nó no grafo da ferramenta *Node-RED* pode realizar um processamento ou pode ser um ponto de entrada/saída para comunicação em um ambiente com serviços na internet (*Facebook*, *Twitter*, *Web Servers*), dispositivos embarcados e sensores, serviços de comunicação (*email*, *telegram*, *SMS*), dentre outros.

O MQTT é uma forma leve e simples que pode ser integrada para comunicação neste ambiente heterogêneo, o qual funciona basicamente como um sistema com publicação (*publish*) e assinatura ou subscrição (*subscribe*), ou seja, uma informação é publicada em um servidor de MQTT que é denominado pelo termo *broker*. A informação é enviada a partir de um dispositivo e/ou servidor e no *broker* esta informação ou mensagem fica disponível para ser capturada por um ou mais dispositivo(es)/servidor(es), a Figura 3 ilustra seu funcionamento básico. A informação é organizada no padrão REST (*Representational State Transfer*) que representa o tópico que foi publicado.

Outra funcionalidade básica e simples do MQTT é o controle da qualidade do serviço ou QoS (*Quality of Service*). A Figura 4 ilustra o seu funcionamento. Há três diferentes níveis de qualidade definidos pelo MQTT, eles variam de acordo com a segurança na entrega da mensagem para o assinante, os níveis são:

- 1) QoS Nível 0: Neste nível, a mensagem publicada não será armazenada por quem a enviou ou reconhecida pelo receptor. Tal comportamento é conhecido, na maioria das vezes, como *"fire and forget"* e provê a mesma garantia que o protocolo TCP (*Transmission Control Protocol*) submetido.
- 2) QoS Nível 1: O nível 1 garante que a mensagem vai ser entregue pelo menos uma vez para o receptor, mas ela também pode ser entregue mais de uma vez.

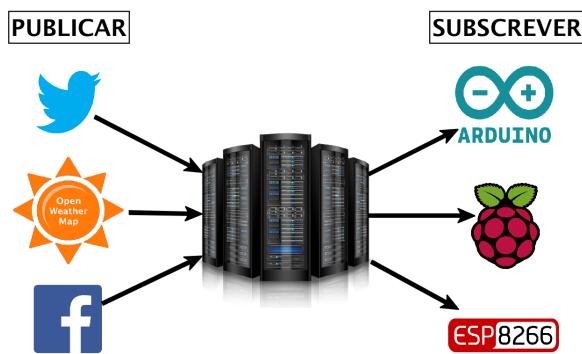


Figura 3. Broker com dispositivos e serviços conectados.

- 3) QoS Nível 2: Este nível garante que a mensagem é recebida apenas uma vez pelo receptor. É o nível mais seguro do MQTT porém o mais lento.

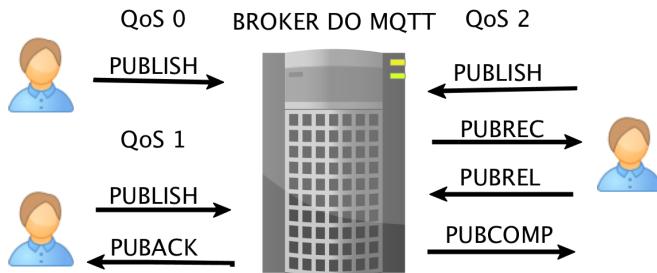


Figura 4. Quality of Service do MQTT

Como mencionado, uma publicação e um ou mais assinantes se comunicam através de tópicos no *broker*. O tópico é uma *string* e seus níveis são separados pelo caractere /, um exemplo de tópico seria universidade/disciplinas/arquiteturadecomputadores. Para se referir aos tópicos, também é permitido o uso de caracteres-curinga, que são o + e o #.

O caractere + é um substituto para um nível de tópico, universidade/+arquiteturadecomputadores poderia se referir a universidade/disciplinas/arquiteturadecomputadores ou a universidade/turmas/arquiteturadecomputadores por exemplo, mas não se referiria a universidade/disciplinas/estruturadedados.

O caractere # é um curinga para vários níveis mas só pode ser usado no final de um tópico, por exemplo, universidade/disciplinas/# poderia se referir a universidade/disciplinas/estruturadedados/turmaA, universidade/disciplinas/estruturadedados/turmaB, universidade/disciplinas/arquiteturadecomputadores/turmaA entre outros.

#### IV. AMBIENTE EXPERIMENTAL

A metodologia proposta aqui é baseada no aprendizado de IoT com exemplos no ambiente *Node-RED*. Inicialmente, a

seção IV-A ilustra exemplos que implementam diversas funcionalidades para explorar o ambiente. A seção IV-B detalha a interface da ferramenta para projetos e a abordagem baseada em fluxos. Enquanto a seção IV-C, apresenta a interface para uso e comunicação, onde usando um ou mais painéis de controle (*dashboard*), o usuário do serviço, através de uma página em um navegador, pode se comunicar com diversos nós do sistema de IoT através de botões, gráficos, etc. A seção IV-D ilustra o potencial de programação dos nós. A seção IV-E enumera várias sugestões de projetos para ensino de IoT. Finalmente a seção IV-F indica onde se pode buscar informação complementar.

##### A. Explorando os recursos

Os primeiros exemplos são simples e ilustram as diversas formas de conectividade. O primeiro exemplo ilustra como conectar: Twitter → Node-RED → MQTT → Arduino/Raspberry, ou seja, como acender um Led em um dispositivo embarcado a partir do Twitter.

A Figura 5 ilustra uma possibilidade de comunicação permitida pelo *Node-RED*. Desta forma, um usuário pode *tweetar* #ligaLED ou #desligaLED. O nó Twitter é uma abstração onde só é necessário preencher algumas informações e todo o serviço de comunicação é configurado de forma transparente. A informação do Twitter é passada para um nó responsável por identificar estas palavras-chave, converter a informação em número, 0 ou 1, ligado ou desligado. Este nó é um exemplo que implementa uma computação que será executada no servidor do *Node-RED*. O código que identifica essa informação é simples e foi escrito em *JavaScript*. A função retorna uma *String* que será repassada para um nó MQTT. Este último nó é capaz de se comunicar com um *broker* MQTT e publicar em um tópico, neste caso o tópico é public/controleLED. Na configuração do nó MQTT, o programador adiciona apenas o endereço do *Broker* e o tópico. A mensagem que será publicada vem do nó anterior. Novamente, uma abstração facilita o uso do protocolo e a comunicação através da rede de forma transparente ao programador.

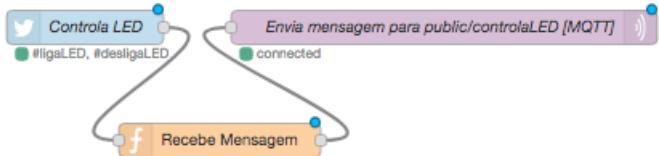


Figura 5. Exemplo de programação baseada em fluxo: Twitter → Node-Red → MQTT → Arduino/Raspberry

O segundo exemplo explora o uso de APIs, para obtenção de dados e acionamento de dispositivos.

Na Figura 6 foi utilizada uma API para obter a previsão do tempo, a configuração dos nós é bem simples, sendo necessário apenas informar a localidade a qual se deseja obter a previsão e a chave da API que pode ser obtida no site do *OpenWeatherMap*. A Figura 7 mostra a janela onde estas informações devem ser inseridas. O nó temperatura

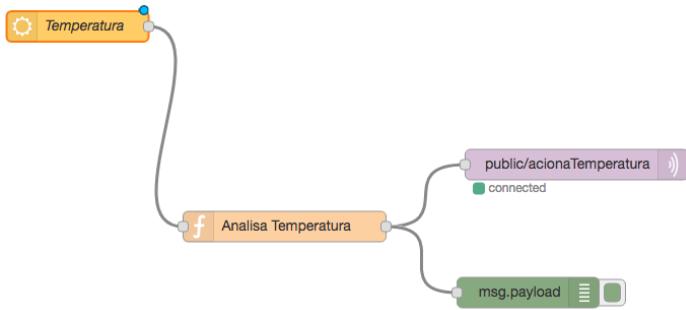
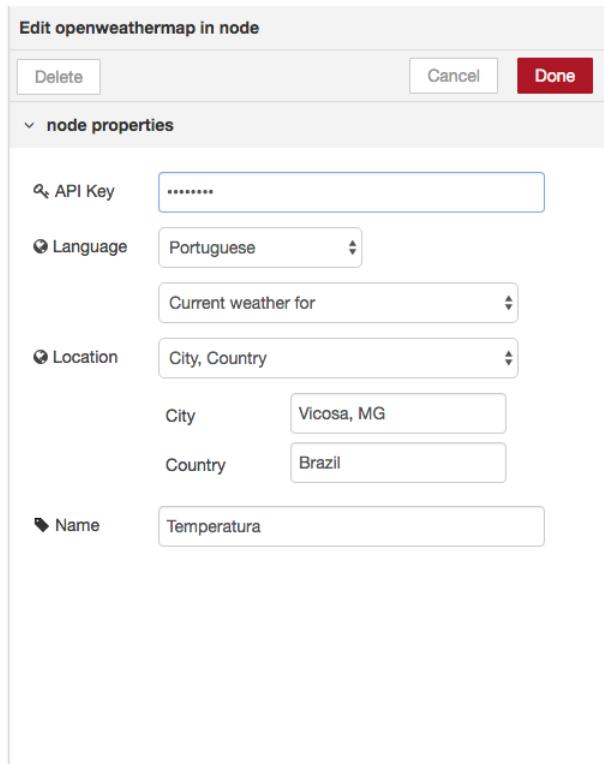


Figura 6. Obter dados de uma API.

retorna a previsão do tempo em JSON que pode ser visto na Figura 8, e este é uma estrutura de dados que pode ser manipulada em *JavaScript*. A partir desta estrutura foi extraído a temperatura máxima utilizando um nó do tipo *function*, no qual é necessário apenas atribuir ao campo *payload* da mensagem a temperatura máxima obtida através do JSON e retornar essa mensagem para o próximo nó. O código tem apenas duas linhas e pode ser facilmente modificado para retirar as outras informações do objeto JSON ilustrado na Figura 8.


 Figura 7. Exemplo de configuração do nó do *OpenWeatherMap*.

No exemplo seguinte foram combinados MQTT e alguns nós especiais como o *Twilio* que permite o envio de SMS.

Na Figura 9 é possível observar que um JSON é obtido a partir do MQTT, deste extrai-se um atributo e seu conteúdo é verificado. Os possíveis valores do atributo são verdadeiro ou

```

    {
        "weather": "Clear",
        "detail": "céu claro",
        "tempk": 299.881,
        "tempc": 26.6,
        "temp_maxc": 26.6,
        "temp_minc": 26.6,
        "humidity": 47,
        "maxtemp": 299.881,
        "mintemp": 299.881,
        "windspeed": 4.37,
        "winddirection": 342.003,
        "location": "Vicosa",
        "sunrise": 1503220180,
        "sunset": 1503261593,
        "clouds": 0,
        "description": "The weather in Vicosa at
coordinates: -20.75, -42.88 is Clear (céu
claro)."
    }

```

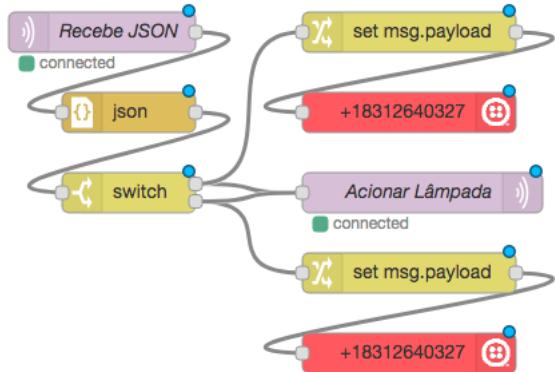
 Figura 8. JSON retornado pelo *OpenWeatherMap*.


Figura 9. MQTT e SMS

falso e, em seguida, utiliza-se um nó *Switch* para criar dois fluxos, um para cada possível valor do atributo, repassando-se assim a informação para o MQTT e um SMS diferente é enviado para cada tópico.

#### B. Painel de Projeto

O painel de projeto do *Node-RED* é dividido em três seções. A primeira mostra os diversos nós disponíveis divididos em *inputs*, *outputs*, *debug* e serviços auxiliares como o *Watson*, Figura 10a. A segunda seção, Figura 10b, é onde o usuário constrói suas aplicações, podendo escolher e conectar os diversos nós disponíveis na plataforma. A terceira seção, Figura 10c, oferece opções de *debug*, porém, para utilizar tal funcionalidade, o usuário deve adicionar os nós próprios para *debug* durante a construção da aplicação.

Também estão disponíveis opções para exportação em JSON que permitem ao usuário levar os chamados *flows* de um *Node-RED* em um servidor para um *Node-RED* em outro servidor, o que é bastante útil quando se deseja manter os *designs* para uso posterior.

## Computer Architecture Education (IJCAE)

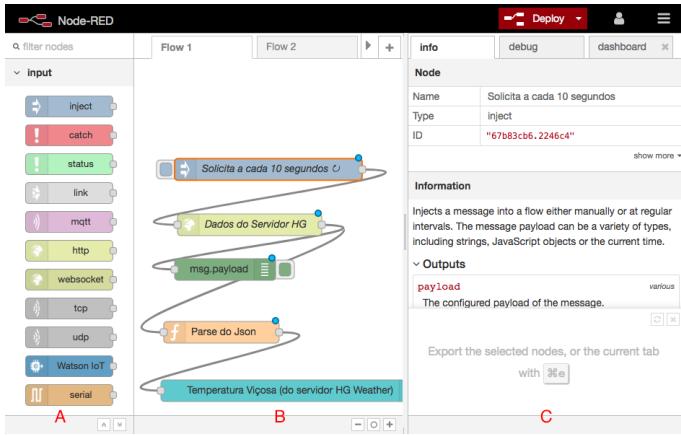


Figura 10. Tela de interface do Node-RED

Além do recurso de exportar, existe a possibilidade de instalar pacotes para incluir mais nós na ferramenta e assim incluir mais recursos. Para realizar a instalação é necessário acessar a opção *Settings* → *Palette* → *Install*, onde é possível pesquisar pelos novos pacotes.

### C. Painel de Controle

Um recurso muito interessante disponível na ferramenta é a possibilidade de construir painéis de controle como os da Figura 11, onde se pode agrupar diversas funcionalidades de controle ou monitoramento em uma interface gráfica amigável. Estão disponíveis diversas funcionalidades, como gráficos de barras, gráficos de pizza, botões, interfaces de entrada de dados, saída de áudio, entre outros.

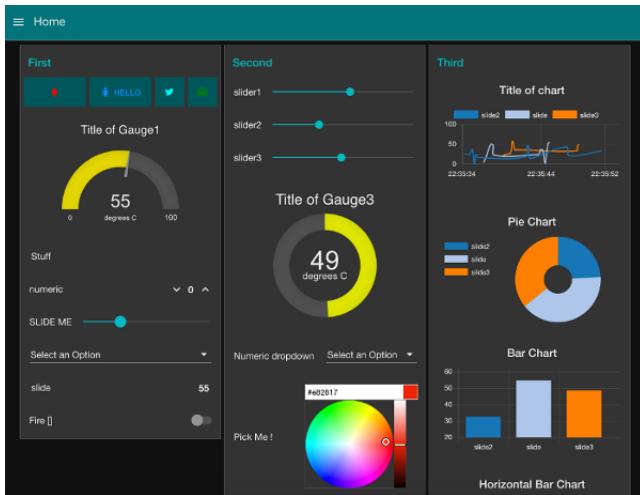


Figura 11. Exemplo com três painéis de controle (*dashboards*) que podem ser acessados por um navegador, mesmo de *smartphones*

Conforme foi mostrado nos exemplos da seção anterior, implementar este recurso é o mesmo que usar o nó desta funcionalidade. Entretanto, para a utilização do *dashboard* é necessário instalar um pacote chamado '*node-red-dashboard*'.

Após a instalação do pacote é criado no domínio de instalação do *Node-RED* uma pasta de "ui", ficando então o

*dashboard* em `dominio-node-red/ui`, onde é possível ter acesso a todos os controles previamente configurados para o uso no painel.

Para o uso da funcionalidade é necessário que se ligue as saídas de dados às saídas do tipo painel. As opções de agrupamento e organização dos *Widgets* (como são chamados os objetos no painel), são mostradas nas opções do nó, facilitando a organização e visualização dos mesmos no resultado final do painel.

### D. Programação

Além da biblioteca de nós que está disponível com mais de mil funcionalidades, existem mais de 600 exemplos de projetos com fluxos disponíveis na biblioteca de exemplos [8]. O programador pode criar novos nós ou simplesmente usar nós que implementam funções para codificar seus algoritmos. O nó mais usado para programação é o nó *function* onde é inserido um código *JavaScript*, no qual ele pode retornar vários valores em saídas diferentes permitindo o controle de fluxo. Existem também nós para programação em *Python* nas versões 2.7 ou 3.0.

O *Node-RED* também pode ser incluído dentro de um código de aplicação ou executar um código no servidor local com uma simples chamada do sistema operacional.

### E. Exemplos de Projetos

Além dos exemplos demonstrados neste artigo, existem diversas outras possibilidades de utilização do *Node-RED*. Com o tutorial encontrado em [8] por exemplo, é possível construir uma API para reconhecimento de imagem usando o *Watson*, o *Node-RED* e uma codificação bastante simplificada.

Há outros projetos que disponibilizam nós e fluxos para serem utilizados sozinhos ou como parte de um projeto maior, a Figura 12 mostra a utilização de um nó encontre meu *iPhone*, onde os dados necessários são apenas o login e a senha da conta da *Apple*. Em adição a estes nós, também é possível encontrar nós que se integram a plataforma *Trello*, que facilitam a utilização das APIs do *Google* e que criam um *ChatBot* para o *Telegram*, *Messenger*, *Facebook* ou *Slack*, entre outros. Também em [8], há um tutorial para integrar o *Amazon Echo* com o *Node-RED* e expandir as habilidades da *Alexa*. O tutorial já fornece um fluxo base pronto mas nada impede a criação de novos comandos adicionais.

A utilização de novos nós em conjunto é possível e um bom exercício para se familiarizar com a ferramenta. Um exercício interessante é criar um *Chatbot* com qual se pode conversar através do *Messenger*, perguntar onde está o meu *iPhone*, quais meus próximos compromissos no *Trello* ou mesmo como está o clima hoje e receber a resposta pelo *Chatbot* no mesmo *Messenger*.

### F. Documentação

Como sugestão para os leitores de material introdutório há o tutorial disponível no site oficial do *Node-RED* [8], sendo ele organizado em lições, no qual a primeira lição apresenta três exemplos: uso de *Twitter* controlando uma saída do *Raspberry*

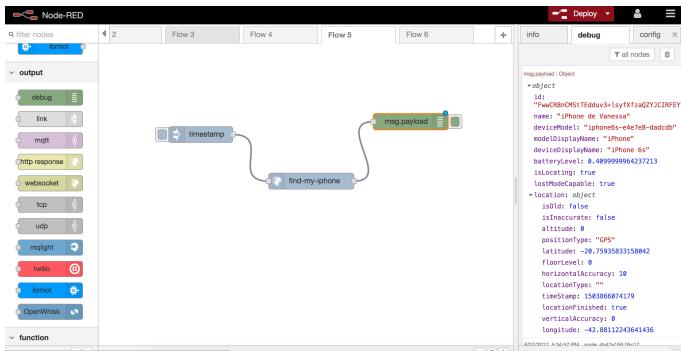


Figura 12. Nô Encontre meu iPhone.

*Pi*, acesso de previsão do tempo com processamento e envio de *email* e como terceiro exemplo um servidor web com apenas seis nós.

São 9 seções no total, 7 já disponíveis, e entre os tópicos é explicado como usar o nó *debug* e o nó de função para abstrair grande parte do código em *JavaScript*. Também é explicado como usar os nós mais comuns, o *MQTT*, *websockets* e requisições *TCP*. É descrito o modelo de programação utilizado pelo *Node-RED*, como usar os nós intermediários e como tirar melhor proveito do *dashboard* oferecido. Além de disponibilizar vários exemplos em cada lição.

## V. CONSIDERAÇÕES FINAIS

Existem diversas opções para ensino e projetos de IoT, dentre os quais este artigo apresenta uma ferramenta da IBM, o *Node-RED* de código aberto. Ao explorar a ferramenta, podemos observar que é flexível e já se encontra em um estado maduro de projeto para ser amplamente utilizada. A Figura 13 mostra uma avaliação no *Google Trends* dos últimos 12 meses relatando as pesquisas relacionadas a cinco ferramentas de IoT, no qual dentre estas o *Node-RED* se destaca. Outro ponto interessante é que pode ser usado para iniciantes e ao mesmo tempo pode ser usado em trabalhos avançados de pesquisa, onde os estudantes podem agregar novas funções ao ambiente. Além de ser código aberto, é disponibilizado documentação, exemplos e uma ampla biblioteca com mais de mil funcionalidades. Acreditamos que o *Node-RED* é uma ferramenta de IoT que irá se popularizar rapidamente com o paradigma de projeto orientado a fluxos. O conceito de nós e de que uma aplicação ou serviço pode ter vários pontos de entrada e saída, desacopla os serviços em fluxos, permitindo uma visão local dos componentes que estão interagindo e modela de forma simples um ambiente complexo onde tudo está potencialmente conectado.

## VI. AGRADECIMENTOS

Os autores agradecem ao CNPq, CAPES e FAPEMIG pelo suporte fornecido no desenvolvimento do trabalho.

## REFERÊNCIAS

- [1] M. Weiser, "The computer for the 21st century," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 3, no. 3, pp. 3–11, Jul. 1999. [Online]. Available: <http://doi.acm.org/10.1145/329124.329126>

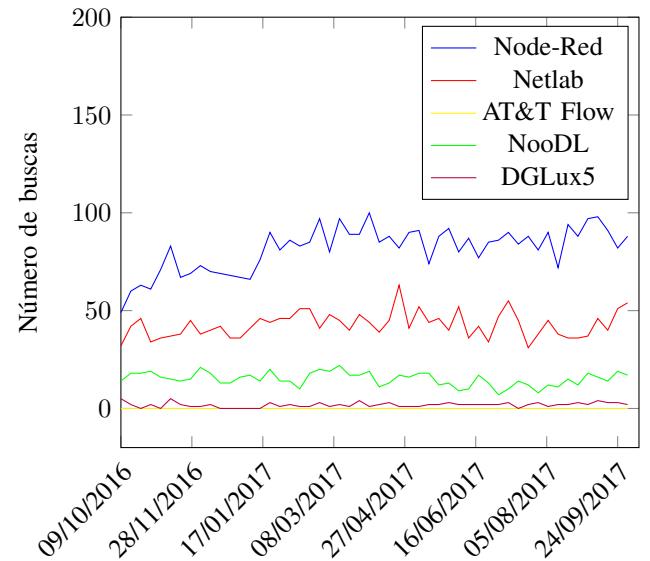


Figura 13. Tendências de Ferramentas de Projeto de IoT

- [2] M. Blackstock and R. Lea, "Toward a distributed data flow platform for the web of things (distributed node-red)," in *Proceedings of the 5th International Workshop on Web of Things*. ACM, 2014, pp. 34–39.
- [3] J. Mineraud, O. Mazhelin, X. Su, and S. Tarkoma, "A gap analysis of internet-of-things platforms," *Computer Communications*, vol. 89, pp. 5–16, 2016.
- [4] R. Kleinfeld, S. Steglich, L. Radzivonowicz, and C. Doukas, "glue. things: a mashup platform for wiring the internet of things with the internet of services," in *Proceedings of the 5th International Workshop on Web of Things*. ACM, 2014, pp. 16–21.
- [5] D. Guinard and V. Trifa, *Building the web of things: with examples in node.js and raspberry pi*. Manning Publications Co., 2016.
- [6] N. K. Giang, M. Blackstock, R. Lea, and V. C. Leung, "Developing iot applications in the fog: A distributed dataflow approach," in *Internet of Things (IOT), 2015 5th International Conference on the*. IEEE, 2015, pp. 155–162.
- [7] K. Kobylinski, J. Bennett, N. Seto, G. Lo, and F. Tucci, "Enterprise application development in the cloud with ibm bluemix," in *Proceedings of 24th Annual International Conference on Computer Science and Software Engineering*. IBM Corp., 2014, pp. 276–279.
- [8] "Coleção de links para o ensino e uso de Node-RED," [http://iot.dpi.ufv.br/links\\_artigo.html](http://iot.dpi.ufv.br/links_artigo.html), accessed: 2017-08-20.
- [9] U. Hunkeler, H. L. Truong, and A. Stanford-Clark, "Mqtt-s—a publish/subscribe protocol for wireless sensor networks," in *Communication systems software and middleware and workshops, 2008. comsware 2008. 3rd international conference on*. IEEE, 2008, pp. 791–798.
- [10] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," *Future generation computer systems*, vol. 29, no. 7, pp. 1645–1660, 2013.
- [11] "A dashboard UI for Node-RED, howpublished = <https://github.com/node-red/node-red-dashboard>, note = Accessed: 2017-08-20."