



**GUILHERME BRUNO ARAÚJO PIMENTA**

**USO DA FERRAMENTA NODE-RED EM PROCESSOS DE  
AUTOMATIZAÇÃO NO CENÁRIO DA QUARTA REVOLUÇÃO  
INDUSTRIAL**

**LAVRAS – MG**

**2019**

**GUILHERME BRUNO ARAÚJO PIMENTA**

**USO DA FERRAMENTA NODE-RED EM PROCESSOS DE AUTOMATIZAÇÃO NO  
CENÁRIO DA QUARTA REVOLUÇÃO INDUSTRIAL**

Monografia apresentada à Universidade Federal de Lavras, como parte das exigências do curso de Engenharia de Controle e Automação, para a obtenção do título de Bacharel.

Prof. Dr. Raphael Winckler de Bettio

Orientador

Prof. Dr. Bruno Henrique Groenner Barbosa

Coorientador

**LAVRAS – MG**

**2019**

**GUILHERME BRUNO ARAÚJO PIMENTA**

**USO DA FERRAMENTA NODE-RED EM PROCESSOS DE AUTOMATIZAÇÃO NO  
CENÁRIO DA QUARTA REVOLUÇÃO INDUSTRIAL**

**NODE-RED APPLIED TO AUTOMATION PROCESSES IN THE FOURTH  
INDUSTRIAL REVOLUTION SCENARIO**

Monografia apresentada à Universidade Federal de Lavras, como parte das exigências do curso de Engenharia de Controle e Automação, para a obtenção do título de Bacharel.

APROVADA em 08 de novembro de 2019.

Prof. Dr. Raphael Winckler de Bettio UFLA  
Prof. Dr. Tadayuki Yanagi Junior UFLA  
Profa. Dra. Jaqueline Severino da Costa UFLA

Prof. Dr. Raphael Winckler de Bettio  
Orientador

Prof. Dr. Bruno Henrique Groenner Barbosa  
Co-Orientador

**LAVRAS – MG  
2019**

*“Our only limitations are those we set up in our own minds.”  
(Napoleon Hill)*

## RESUMO

Com o advento da quarta revolução industrial, a população é exposta a uma grande quantidade de novas tecnologias, que abrangem não somente a indústria, como também diversos setores da sociedade. Este trabalho apresenta um levantamento das principais tecnologias associadas a quarta revolução industrial, mostrando como elas permeiam algumas áreas da sociedade e como uma plataforma de programação em fluxo orientada a *Internet* das Coisas pode atuar em diferentes processos de automatização dentro desse cenário. Foram desenvolvidas aplicações por meio do uso da ferramenta *Node-Red*, expondo suas características e comportamentos para situações nas áreas agrícola, industrial e residencial. Nesse trabalho é apresentada a programação realizada em cada um desses ambientes, no qual foram criados exemplos como: gerenciador para irrigação automatizada e inteligente, controle de qualidade em uma linha de produção e automação residencial. De modo a realizar a validação do funcionamento de cada aplicação, criou-se telas de teste, nas quais permitiram a inserção de dados que simularam leituras de sensores ou comportamentos específicos do sistema. Ao final do processo foram listadas algumas das principais vantagens e desvantagens observadas quanto ao uso do *Node-Red* para as aplicações em questão. Por fim, foi verificado a importância desse tipo de abordagem frente a dinâmica trazida pela quarta revolução industrial, e como essa ferramenta de baixa complexidade consegue entregar bons resultados, apresentando assim, uma solução viável dentro do paradigma de programação da quarta revolução industrial.

**Palavras-chave:** Node-Red. Quarta revolução industrial. *Internet* das coisas. Programação em fluxo. Indústria 4.0.

## ABSTRACT

The advent of the fourth industrial revolution, exposed to the population a host of new technologies, encompassing not only the industry field but also various sectors of society. This paper presents a study of the key technologies associated with the fourth industrial revolution, showing how they permeate some areas of society and how an IoT oriented flow-based programming platform can be used on different automation processes in this scenario. Applications were developed using Node-Red, exposing its characteristics and behaviors for situations in the agricultural, industrial and residential areas, presenting the programming behind in each of these environments. As a result were create the following applications: automated irrigation system, production quality control and home automation. In order to validate the operation of each application, virtual environments were created, which allowed the insertion of data that simulated sensor readings or specific behaviors of the system. At the end were listed some of the main advantages and disadvantages observed regarding the use of Node-Red for these applications. Finally, it was verified the importance of this approach in face of the dynamics brought by the fourth industrial revolution, and how this low complexity tool can deliver good results, thus presenting a viable solution within the programming paradigm of the fourth industrial revolution.

**Keywords:** Node-Red. Fourth industrial revolution. Internet of things. Flow-based programming. Industry 4.0.

## SUMÁRIO

1	<b>Introdução</b>	7
1.1	<b>Objetivos</b>	8
1.2	<b>Estrutura do trabalho</b>	8
2	<b>Desenvolvimento</b>	9
2.1	<b>Referencial teórico</b>	9
2.1.1	<b>A evolução da indústria</b>	9
2.1.1.1	<b>Primeira revolução industrial</b>	9
2.1.1.2	<b>Segunda revolução industrial</b>	10
2.1.1.3	<b>Terceira revolução industrial</b>	11
2.1.1.4	<b>Quarta revolução industrial</b>	11
2.1.2	<b>Tecnologias presentes na quarta revolução industrial</b>	13
2.1.2.1	<b>Automação</b>	13
2.1.2.2	<b>Inteligência artificial</b>	14
2.1.2.3	<b>Robótica</b>	15
2.1.2.4	<i>Internet das coisas</i>	15
2.1.2.5	<b>Visão computacional</b>	16
2.1.2.6	<i>Deep learning</i>	17
2.1.2.7	<i>Big Data</i>	18
2.1.3	<b>Ferramentas presentes na quarta revolução industrial</b>	19
2.1.3.1	<i>Node-Red</i>	20
2.1.3.2	<b>Raspberry PI</b>	22
2.1.3.3	<i>Chip ESP8266</i>	23
2.1.3.4	<b>Protocolo MQTT</b>	24
2.2	<b>Metodologia</b>	26
2.2.1	<b>Importância do <i>Node-Red</i> no cenário atual</b>	26
2.2.2	<b>Exemplo de aplicação do <i>Node-Red</i> no setor agrícola</b>	28
2.2.2.1	<b>Projeto</b>	28
2.2.2.2	<b>Ambiente de operação</b>	28
2.2.2.3	<b>Ambiente de desenvolvimento</b>	30
2.2.2.4	<b>Resultados e simulação</b>	35
2.2.3	<b>Exemplo de aplicação do <i>Node-Red</i> no setor industrial</b>	37

2.2.3.1	Projeto . . . . .	37
2.2.3.2	Ambiente de operação . . . . .	38
2.2.3.3	Ambiente de desenvolvimento . . . . .	39
2.2.3.4	Resultados e simulação . . . . .	43
2.2.4	Exemplo de aplicação do <i>Node-Red</i> na área da domótica . . . . .	45
2.2.4.1	Projeto . . . . .	45
2.2.4.2	Ambiente de operação . . . . .	46
2.2.4.3	Ambiente de desenvolvimento . . . . .	47
2.2.4.4	Resultados e simulação . . . . .	52
2.2.5	Implementação em <i>hardware</i> . . . . .	52
2.3	Resultados e discussão . . . . .	55
3	Conclusão . . . . .	57
	REFERÊNCIAS . . . . .	58



## 1 INTRODUÇÃO

A sociedade já presenciou grandes revoluções ao longo da história que mudaram completamente o modo como os bens de consumo são produzidos, resultando em notáveis reestruturações não apenas na indústria como também no modo de vida das pessoas. Atualmente, se configura o início de uma grande revolução tecnológica que vem causando disruptura em toda indústria e no dia a dia da população, a chamada quarta revolução industrial.

Essa nova era é marcada pelo uso massivo de tecnologias recém descobertas, ou pelo aperfeiçoamento de técnicas já conhecidas que só se tornaram possíveis graças ao poder de processamento dos computadores modernos e aos avanços de outras áreas correlacionadas, como exemplo observa-se o uso da automação, inteligência artificial, *Internet* das coisas, visão computacional, dentre outras. Todos esses recursos unidos vêm alterando a maneira como os produtos são fabricados e como os serviços são prestados ao consumidor final.

Embora grande parte das tecnologias relacionadas a quarta revolução industrial tenha surgido direcionada ao processo fabril, elas não se limitam a esse segmento, sendo que, já é evidenciado seu uso nas áreas de educação, medicina, agricultura, mobilidade, comércio, dentre vários outras. Esse uso diversificado que abrange numerosas esferas de aplicações é uma das características que faz da quarta revolução industrial um período singular.

Diversos setores da sociedade já estão fazendo uso das ferramentas e tecnologias associadas a quarta revolução industrial, obtendo significativos resultados e novas possibilidades e formas de negócios, tornando seus produtos e serviços melhores e mais competitivos no mercado. Até mesmo segmentos que possuem maior rigor em quesitos de segurança e padrão de fabricação como o setor aeronáutico, também vem adotando grande parte dessas novas tecnologias e as incorporando em seus processos.

Com o novo paradigma criado pela quarta revolução industrial, novas ferramentas de programação e integração entre sistemas são desenvolvidas, de modo a obter o máximo possível dos recursos trazidos por esse novo modelo. Dessa forma, ferramentas não convencionais usadas para a automação de processos passam a ganhar notoriedade e importância, devido a sua facilidade de uso e capacidade de integrar sistemas pertencentes a esse novo cenário tecnológico.

Atualmente é possível encontrar ferramentas gratuitas e de fácil aprendizado destinadas ao desenvolvimento de sistemas de automatização e controle, seja de um processo fabril,

agrícola ou até mesmo residencial. Esse tipo de ferramenta apresenta grande importância em processo de automatização, sendo assim o foco de estudo desse trabalho.

A disruptura trazida pela quarta revolução industrial tende a se tornar maior a cada dia, conseqüentemente, novas ferramentas tecnológicas são criadas, surgem novos modelos de negócios, e novas profissões são geradas. Sendo assim, se faz necessária a análise de uma nova forma de automatizar processos, que seja capaz de integrar variadas fontes de dados, bem como ser de fácil implementação e baixo custo.

## 1.1 Objetivos

Esse trabalho tem o objetivo de exemplificar como a linguagem de fluxo orientada a *Internet* das coisas pode atuar em processos de automatização dentro do cenário que envolve as tecnologias trazidas pela quarta revolução industrial. Apresentando por meio de simulações e implementação em *hardware* exemplos de aplicações nas áreas agrícola, industrial e residencial, utilizando para tal a ferramenta *Node-Red*, expondo assim, suas características e importância dentro desse novo paradigma.

## 1.2 Estrutura do trabalho

Esse trabalho foi estruturado em cinco partes, sendo elas:

- Primeira parte: breve histórico a respeito das revoluções industriais anteriores, evidenciando quais foram as principais tecnologias que marcaram cada uma delas;
- Segunda parte: apresentação de tecnologias e ferramentas associadas à quarta revolução industrial;
- Terceira parte: Análise da importância da ferramenta *Node-Red* no cenário tecnológico atual;
- Quarta parte: aplicação da ferramenta *Node-Red* em processos de automatização nos setores agrícola, industrial e residencial;
- Quinta parte: conclusão a respeito do estudo realizado.

## **2 DESENVOLVIMENTO**

### **2.1 Referencial teórico**

#### **2.1.1 A evolução da indústria**

Durante anos a indústria vem se transformando, passando por um conjunto de mudanças que não apenas alterou a forma de produção de bens de consumo, mas também toda a dinâmica social já estabelecida em cada época. A revolução industrial se apresenta como uma das raras ocasiões na história do mundo no qual a espécie humana alterou sua estrutura de existência. De fato, o único desenvolvimento anterior comparável em termos de grande magnitude foi a revolução Neolítica, na qual ocorreu a conversão da caça e coleta para a agricultura como forma básica de produção para a sobrevivência (STEARNS, 2012).

Cada revolução industrial foi marcada por um salto tecnológico atribuído a algum material, máquina ou modo de produção. A organização da sociedade atual e o modelo de vida adotado é em grande parte devido as mudanças trazidas pelo desenvolvimento ocorrido nessas revoluções tecnológicas.

As revoluções industriais são majoritariamente classificadas em três fases, cada qual com suas tecnologias e peculiaridades. A seguir é relatado resumidamente cada revolução industrial, com os principais fatores que contribuíram para a brusca mudança do cenário vigente da época.

##### **2.1.1.1 Primeira revolução industrial**

Essa fase se estabeleceu primeiramente na Inglaterra entre os séculos XVIII e XIX, se espalhando rapidamente para outros países. Tal revolução se deu em um contexto marcado pela atividade produtiva manual e artesanal com pouco uso de ferramentas mecânicas. Nessa época, em diversos processos produtivos o artesão tinha total controle e conhecimento de todas as etapas que envolvia a produção do bem, participando ativamente desde a obtenção da matéria prima até a comercialização do bem (DEANE, 1979).

A primeira revolução foi marcada principalmente pela descoberta da utilidade do carvão mineral como fonte energética, e do aperfeiçoamento da máquina a vapor por James Watt e Matthew Boulton (THURSTON, 1878), tornando assim possível o desenvolvimento de novas máquinas e ferramentas capazes de realizar tarefas antes feitas manualmente. Com isso, grandes

mudanças ocorreram sobretudo no setor têxtil com a inserção do tear mecânico concebido por Edmund Cartwright (RIELLO, 2013).

Com a introdução dessas novas descobertas na sociedade, ocorreram mudanças significativas não apenas no setor produtivo, mas como também no transporte com as locomotivas a vapor, alterando completamente o modelo estabelecido na época.

Segundo Deane (1979), a primeira revolução industrial teve como algumas de suas conquistas o aumento da quantidade de bens de consumo disponível para o consumidor médio, bem como um alongamento da cadeia entre o produtor e o consumidor final. Contudo, devido ao elevado crescimento do número de habitantes nas cidades, diversos problemas sociais também surgiram como consequência dessa revolução.

### **2.1.1.2 Segunda revolução industrial**

A segunda revolução industrial ocorreu entre os séculos XIX e XX, principalmente na Inglaterra, França, Alemanha e Estados Unidos, marcada por transformações nos setores de comunicação, transporte e manufatura (OUTMAN; OUTMAN, 2003).

Durante essa fase foi evidenciado um aperfeiçoamento das ferramentas já existentes no século XVIII, tal como a introdução de novas descobertas tecnológicas conforme tratado em (SMIL, 2005), tendo como principais, o uso do combustível fóssil e da eletricidade como fontes de energia, o desenvolvimento de meios de comunicação como o telégrafo, novos processos de fabricação de aço e o motor a combustão, essas foram algumas das inovações presentes na segunda fase da revolução industrial que remodelou a época.

Cada descoberta dessa fase teve sua importância e contribuiu fortemente no progresso da sociedade, principalmente o desenvolvimento da eletricidade, que deu suporte a várias outras invenções, como a eletrônica, dispositivos eletromecânicos, sistemas de iluminação, dentre outras.

Esse período foi caracterizado também pela produção em massa de bens de consumo, com a introdução das linhas de montagem e do modelo *fordista* de produção. No qual é possível notar que o trabalho se tornou altamente repetitivo e estressante, com o ritmo ditado por uma esteira, onde o trabalhador é considerado apenas mais uma peça que compõe a organização fabril.

A segunda revolução industrial foi sem dúvida um período de grandes descobertas e mudanças em diversos ramos da sociedade, contribuindo também fortemente nos setores da agricultura, medicina e na indústria química, introduzindo a sociedade a uma nova era.

### **2.1.1.3 Terceira revolução industrial**

Nos anos de 1950 aos anos 2000 a indústria foi contemplada com o advento da eletrônica, que deu suporte ao desenvolvimento da computação e da automação. Nessa época, devido aos satélites e a *internet*, o volume e a velocidade das informações trocadas sofreram um aumento considerável, o que contribuiu ainda mais para a disseminação da industrialização ao redor do mundo (STEARNS, 2012).

A diversificação das fontes de energia foi um marco desse período, além do carvão mineral e do petróleo que foram utilizados massivamente nas revoluções anteriores, as energias hidrelétrica, nuclear e eólica passam a fazer parte da nova matriz energética.

A evolução da eletrônica permitiu a redução das dimensões físicas dos computadores, bem como o aumento de sua capacidade de processamento e armazenamento de dados, o que contribuiu para a implementação massiva da automação nas indústrias, principalmente nas linhas de montagem, agindo na substituição da mão de obra humana em tarefas de natureza repetitiva e de baixa complexidade intelectual, gerando assim grande realocação de postos de trabalho (RIFKIN, 2011).

De mesma forma que suas predecessoras, essa revolução adicionou nova dinâmica ao modo de produção de bens. O *Toyotismo*, foi um modelo incorporado em algumas indústrias, na qual a flexibilização da produção e a busca por mão de obra especializada eram características presentes nesse modelo, estando assim mais adequado para o ambiente globalizado do momento.

### **2.1.1.4 Quarta revolução industrial**

Desde meados do século XVIII houveram revoluções que mudaram o paradigma da sociedade vigente, a primeira com a energia a vapor e a mecanização, a segunda tornou possível a produção em massa devido ao advento da energia elétrica, a terceira com a introdução da computação e robótica em virtude do desenvolvimento da eletrônica, agora configura-se o início de uma quarta revolução industrial.

Segundo Schwab (2017), a quarta revolução industrial é caracterizada por uma *internet* mais móvel e onipresente, por sensores menores e mais baratos e pelo uso massivo de inteligência artificial e aprendizado de máquina, dando suporte as *Smart factories* e permitindo a criação de novos modelos de operação. Além dessas tecnologias, a quarta revolução industrial também engloba outras áreas, como sequenciamento genético, nanotecnologia, computação quântica e várias outras. Essa interação entre o ambiente físico, digital e biológico é o que marca essa revolução e a diferencia de suas antecessoras.

Atualmente já é possível encontrar uma quantidade considerável de novas empresas que fazem uso das tecnologias associadas à quarta revolução industrial, e que trabalham de forma diferente do padrão tradicional, ou seja, que causaram disruptura nos seguimentos atuantes, conseguindo assim, criar novos modelos de mercado e obter grandes lucros a custo marginal próximo a zero, segundo Ismail et al. (2014), essas são as chamadas “Organizações Exponenciais”.

Devido ao grande potencial de impacto que quarta revolução industrial pode causar na realidade atual, diversos setores da sociedade estão discutindo e traçando planos ao redor das mudanças que tais tecnologias trazem e podem trazer para os próximos anos. Esse é um assunto que vem ganhando bastante visibilidade, sendo discutido principalmente no Fórum Mundial de Economia, no qual é reunido diversas autoridades e especialistas das áreas, a fim de debaterem a respeito do uso dessas tecnologias e de seus impactos.

A figura 2.1 ilustra de forma resumida algumas das tecnologias que tiveram fundamental importância em cada revolução industrial e que serviram de base para que o desenvolvimento pudesse ocorrer até o momento atual.

Figura 2.1 – Revoluções industriais



Fonte: do autor

## 2.1.2 Tecnologias presentes na quarta revolução industrial

### 2.1.2.1 Automação

Segundo Lamb (2013), automação corresponde ao uso de lógica de comandos e mecanização de equipamentos para substituir decisões e atuações manuais realizadas em atividades exercidas por humanos. Acredita-se que o termo foi primeiramente usado em 1940 na fábrica *Ford Motor* para descrever ações e controles realizados por máquinas.

A automação não se restringe em apenas transformar uma tarefa manual em automática, por meio de um conjunto de técnicas ela consegue melhorar a qualidade dos produtos, reduzir custos, aumentar a produtividade, etc. No setor industrial, uma empresa que não faz uso de meios de automação, dificilmente consegue se manter competitiva no mercado.

Embora processos automatizados se apresentam em maior escala no meio fabril, esses também passaram a estar presentes em outras áreas, como nas residências (domótica), na qual é realizada o sensoriamento e automatização de equipamentos domésticos, a fim de fornecer maior conforto e segurança aos moradores; no setor de prestação de serviços (automação comercial), com processos automáticos de controle de estoque, fornecedores, clientes, etc. Outro elemento que está cada vez mais fazendo uso das ferramentas de automação são as pessoas, portando dispositivos integrados ao corpo humano, de forma a promover melhor monitoramento da saúde, bem como auxiliar na execução de tarefas e atividades físicas, como exemplo se observa o uso dos relógios inteligentes.

A automação pode ser classificada em níveis (figura 2.2), cada qual implementa determinados tipos de tecnologias e são destinados e operados por diferentes setores de uma empresa. Com o advento da Indústria 4.0 esses níveis estão sofrendo alterações, ficando cada vez mais interligados.

Figura 2.2 – Pirâmide da automação



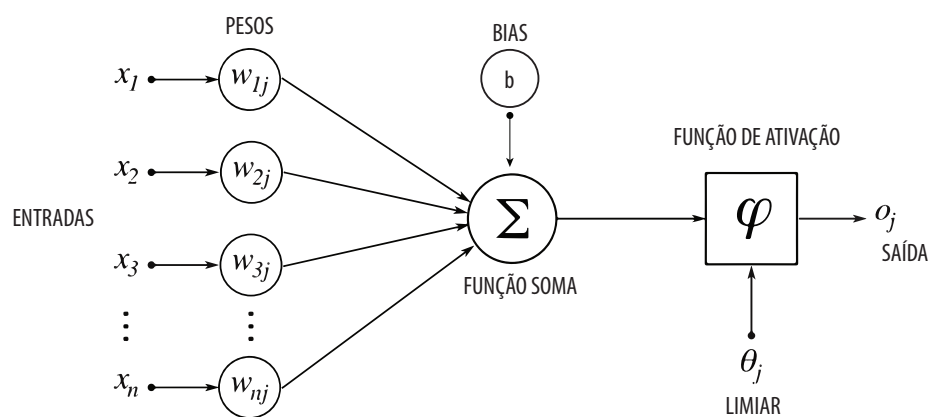
Fonte: do autor

### 2.1.2.2 Inteligência artificial

Existem diversas definições para inteligência artificial, que dependem tanto da área em que ela é analisada quanto da época de estudo. Segundo a definição proposta por Kurzweil (1990), inteligência artificial corresponde a arte de criar máquinas que desempenham funções que exigiriam inteligência quando essas fossem realizadas por pessoas.

Embora o termo esteja bem difundido na atualidade, os primeiros passos da inteligência artificial se estabeleceram em 1943 com a publicação de um artigo por Warren McCulloch e Walter Pitts (MCCULLOCH, 1943), no qual é apresentado pela primeira vez um modelo matemático que simula uma estrutura neural. Em 1958 Roseblatt inventou o *Perceptron*, um algoritmo computacional que simula o neurônio, sendo este usado para a classificação binária. A Figura 2.3 ilustra a representação do neurônio artificial.

Figura 2.3 – Neurônio artificial



Fonte: do autor

De posse do poder de processamento dos computadores modernos, principalmente pelo emprego de GPUs<sup>1</sup>, a inteligência artificial foi levada a outro patamar, podendo agora fazer uso de técnicas e algoritmos que envolvam *Deep learning*<sup>2</sup> de maneira mais rápida, tornando possível a resolução de problemas mais complexos e abstratos.

Atualmente a inteligência artificial vem atuando em inúmeras áreas do conhecimento, seja no setor da saúde (auxiliando em diagnósticos médicos), no reconhecimento da fala e traduções de idiomas, ou até mesmo em carros autônomos, que não seria uma realidade sem os ganhos que essa tecnologia proporciona.

<sup>1</sup> Graphics Processing Unit: consiste em um chip utilizado para processamento gráfico, apresentando ótima performance em cálculos envolvendo matrizes.

<sup>2</sup> Redes neurais profundas: grandes conjuntos de camadas de neurônios artificiais, utilizado principalmente para modelar problemas de alta complexidade.



### 2.1.2.3 Robótica

Robótica corresponde a interligação de diversas áreas da ciência, a fim de desenvolver dispositivos eletromecânicos capazes de realizar tarefas que seriam a princípio atribuídas a humanos (FRANCESCHETTI, 2018). Nos dias de hoje a robótica já está presente em quase todos os setores da sociedade, seja no campo, na indústria, no comércio ou nas residências.

Existem diversos tipos de robôs realizando as mais variadas atividades, como exemplo: os braços robóticos na indústria, desempenhando atividades como soldagem e montagem; máquinas colhedoras no campo, ajudando na colheita e separação do alimento; assistentes virtuais no comércio, substituindo o atendimento humano tradicional; protótipos de nano-robôs na medicina, usados para o combate a enfermidades a nível molecular; dentre diversos outros exemplos apresentados por Staff et al. (2018).

Na indústria, a robótica teve papel fundamental no desenvolvimento produtivo, as linhas de montagem não teriam o mesmo rendimento sem a introdução das ferramentas robotizadas, tendo essas a capacidade de atuar no auxílio da atividade ou até mesmo realizar sua completa execução quando a natureza da tarefa envolvia grande risco se fosse atribuída a uma pessoa.

Atualmente com a evolução da robótica, observa-se a introdução de robôs colaborativos na indústria, ou seja, robôs que podem trabalhar lado a lado do ser humano sem oferecer riscos a segurança desses, auxiliando-o na execução de suas atividades.

A robótica sempre se preocupou em observar a natureza e tentar reproduzir seu funcionamento com os conhecimentos e aparatos disponíveis, uma vez que os seres vivos são produto de anos de um processo evolutivo, uma máquina construída de modo a simular sua biologia e comportamento tende a ser bem sucedida, principalmente com a miniaturização da eletrônica e da incorporação de técnicas de inteligência artificial, configurando assim uma nova geração de robôs que vem demonstrando um desempenho sem precedentes.

### 2.1.2.4 Internet das coisas

*Internet of things (IOT)* ou “*Internet das coisas*”, refere-se à incorporação de eletrônica embarcada em diversos objetos, tornando esses aptos a trocarem informações com outros equipamentos, fazendo assim parte de uma rede de dispositivos conectados entre si. Quando objetos e máquinas passam a ser sensoreados e controlados através da *internet*, uma integração entre o mundo físico e virtual é estabelecida, possibilitando diversas melhorias no que diz respeito a eficiência, precisão, automação e planejamento (HANES et al., 2017).

Dispositivos IOTs se tornaram possíveis graças a evolução da eletrônica embarcada e dos meios de comunicação, encontra-se no mercado uma variedade de componentes eletrônicos que já seguem o paradigma da *Internet* das coisas. Um bom exemplo desse tipo de dispositivo é o módulo de baixo custo ESP8266<sup>3</sup>, que pode ser usado tanto para adaptar aparelhos convencionais aos padrões IOT como na criação de novos equipamentos.

Diversas empresas já fabricam seus produtos com capacidade de comunicação e integração com a *internet*, seja na indústria com as próprias máquinas gerando dados de utilização e desempenho, quanto nas residências com o uso de eletrodomésticos programáveis, câmeras IP, fechaduras eletrônicas, assistentes virtuais, etc. Outro setor que vem fazendo grande uso do IOT é o setor médico, com ferramentas direcionadas ao monitoramento de sinais vitais, auxiliando assim no acompanhamento médico e análise dos efeitos de medicamentos sobre os pacientes.

Uma questão que gera barreiras quanto a implementação maior da tecnologia, se diz respeito as questões de segurança e privacidade, uma vez que dispositivos IOT estão conectados a rede, esses se não configurados da forma correta podem ser alvos de ataques ou coleta de dados não autorizada, o que pode causar não apenas danos materiais como morais.

Conforme citado por Tripathy e Anuradha (2017), na implementação do IOT a privacidade deve ser fortemente considerada, embora a noção de privacidade vem se alterando com o tempo, como o exemplo das câmeras de vigilância que antes eram vistas invasivas, mas hoje já se tornaram comum. Ainda que essa tecnologia ofereça significativos benefícios tanto à indústria como aos usuários, seus impactos na sociedade devem ser considerados e estudados, de modo que informações sensíveis não sejam expostas a terceiros sem o consentimento do usuário.

#### **2.1.2.5 Visão computacional**

Visão computacional corresponde a um ramo dentro da ciência, que trabalha criando formas artificiais de se adquirir informações a partir de imagens. Segundo Sonka, Hlavac e Boyle (2014), essa tecnologia se trata de maneira a replicar a percepção e o entendimento da visão humana por meio de sistemas eletrônicos, tornando possível a extração de informações de interesse.

---

<sup>3</sup> ESP8266: chip de baixo custo produzido pela Espressif Systems que integra um microcontrolador com comunicação Wi-fi e implementação do protocolo TCP/IP.

De modo geral, o sistema de visão computacional pode ser subdividido em 5 etapas, sendo elas: aquisição, estágio onde o arquivo digital é produzido por meio da conversão das grandezas físicas captadas pelos sensores em valores digitais; pré-processamento, etapa de preparação da imagem a fim de eliminar características de não interesse, de modo a tornar os estágios posteriores mais eficientes; extração de características, fase na qual é aplicado operações matemáticas com o intuito de identificar determinada característica em uma imagem; detecção e segmentação, nessa etapa o objeto de interesse é detectado e separado do restante da imagem; processamento, estágio final onde é realizada a classificação da informação de desejo, fornecendo dados relevantes para a tomada de decisão.

Com a evolução das placas gráficas e das técnicas de processamento de imagem, a visão computacional tem ganhado cada vez mais espaço em diversos setores, sua aplicação pode ser observada por exemplo na área médica, auxiliando em diagnósticos; em processos de fabricação, trabalhando no controle de qualidade por meio de detecção de irregularidades visualmente observáveis, dentre outros. Nos últimos anos constatou-se seu uso em veículos autônomos, no qual a análise e a interpretação dos arredores do veículo se estabelecem em grande parte por essa tecnologia, sendo vista como uma alternativa ao uso do LIDAR<sup>4</sup>.

Atualmente existem diversos algoritmos e técnicas computacionais capazes de atuar em variados cenários, dentre as inúmeras abordagens apresentadas por Davies (2017), é mostrado a incorporação de aprendizado de máquina nos processos de reconhecimento e detecção de objetos, técnica que vem obtendo maior precisão e confiabilidade devido a evolução das GPUs<sup>5</sup> e do aumento do banco de dados disponível para treinamento das redes neurais profundas.

#### **2.1.2.6 *Deep learning***

*Deep learning* ou aprendizado profundo é um ramo do aprendizado de máquina que tem como base a matemática, a neurociência e a computação. Embora sua origem data de 1950, com o advento da inteligência artificial e devido a evoluções no poder computacional das máquinas, essa se popularizou nos últimos anos. Essa técnica trabalha com base em grandes quantidades de dados, conseguindo assim gradativamente adquirir habilidades para desenvolver determinada tarefa, semelhante ao aprendizado que ocorre em bebês (SEJNOWSKI, 2018).

---

<sup>4</sup> *LIDAR: Light Detection And Ranging*, dispositivo utilizado para mapear objetos ao redor por meio de laser infravermelho, criando assim nuvens de pontos que representam o entorno.

<sup>5</sup> *Graphics processing unit*: placa eletrônica dedicada a processar gráficos em computadores.

De modo a se obter resultados satisfatórios usando redes neurais profundas em cenários complexos, faz-se necessário uma grande quantidade de dados de treinamento, assim, expondo o algoritmo computacional a um banco de dados maior e mais variado, esse pode determinar de forma mais precisa as relações matemáticas que existem entre as informações fornecidas e o objetivo a ser alcançado.

Atualmente *Deep learning* está sendo usado em diversos ramos, dentre eles: reconhecimento facial, hoje em dia uma máquina já consegue identificar pessoas com uma precisão semelhante à de um ser humano; traduções de idiomas, plataformas e *softwares* de tradução estão migrando seu sistema de tradução que era feito apenas com base em banco de dicionários para essa nova ferramenta, que permite melhores resultados e traduções mais naturais; veículos autônomos, a interpretação do ambiente ao redor do veículo, a identificação de pedestres, placas, faixas e outros elementos se dá por meio da visão computacional em conjunto com o uso de redes neurais profundas, permitindo a tomada de decisão automática.

Existem uma variedade de técnicas, arquiteturas e métodos de se implementar uma rede neural profunda, conforme reportado em Goodfellow, Bengio e Courville (2016). O tipo de treinamento, a arquitetura, o número de camadas e neurônios utilizados devem ser projetados de acordo com a natureza do problema, porém em muitas situações, alguns desses parâmetros são obtidos de forma experimental, por meio de tentativa e erro na busca da melhor performance. Em determinados problemas não se sabe ao certo como a máquina realiza a classificação, ou seja, quais informações ela considerou mais que outras e quais ela descartou, como no caso de reconhecimento facial, sendo essa uma das características interessantes dessa ferramenta, o fato de atribuir ao algoritmo a tarefa de descobrir quais os melhores atributos a se considerar em um certo banco de dados.

### **2.1.2.7 Big Data**

Na era da informação atual, dados são gerados a todo momento e possuem grandes valores quando usados da forma correta. Segundo Marr (2016), *Big Data* refere-se ao fato de coletar e analisar grande volume de dados. Isso só se tornou possível devido a evolução dos equipamentos de armazenamento e processamento de dados, bem como a maior quantidade de informações disponíveis, sendo essas introduzidas pela sociedade que se observa mais moderna e dependente dos recursos digitais para a execução de seus trabalhos e tarefas do dia a dia.

Com a *internet* e os aplicativos móveis cada vez mais presentes na sociedade, o volume de dados gerados tem alcançado patamares sem precedentes. A maioria dos serviços digitais utilizados realizam algum tipo de coleta de dados, como: histórico de localização do usuário, gostos, orientação política, produtos comprados, etc. Todos esses dados serão processados a fim de se obter algum tipo de ganho, como: propagandas direcionadas, influência política ou desenvolvimento de alguma estratégia ou plano de ação.

*Big Data* tem sido usado massivamente em aprendizado de máquina, ou seja, uma enorme quantidade de dados são usados para alimentar redes neurais de modo que o algoritmo computacional consiga encontrar as relações de interesse. Isso vem sendo a base para o funcionamento otimizado de *softwares* de reconhecimento por imagem, auxiliando em aplicações como: análises médicas, veículos autônomos, dentre outras. Na indústria atual, diversos fabricantes de equipamentos já estão projetando seus produtos com capacidade de geração de dados de seu próprio funcionamento, permitindo ao administrador realizar a análise dos dados e fazer as devidas projeções.

Conforme apontado por Kitchin (2014), os dados gerados possuem diversos fins dentro das instituições. A proposta do uso desses se fundamenta em construir uma sociedade mais segura, produtiva e eficiente, ou seja, melhorar a condição de vida. Contudo existem preocupações entre usar os dados de forma emancipatória, e usar os dados para seus próprios fins em detrimento de outros. Talvez o gerador da informação seja prejudicado pelos seus próprios dados, mesmo que não tenha feito nenhum ato ilegal. Portanto, a privacidade e o destino final da análise de todas as informações geradas devem ser tratadas de forma adequada, para que os direitos de cada cidadão sejam respeitados ao passo que a melhora dos serviços sejam alcançadas.

### **2.1.3 Ferramentas presentes na quarta revolução industrial**

Atualmente observa-se variadas ferramentas capazes de atuarem em cenários que envolvam as tecnologias associadas à quarta revolução industrial, fornecendo assim, suporte de *hardware* e *software* em aplicações dessa natureza. A seguir foram levantadas algumas das ferramentas que apresentam tais características.

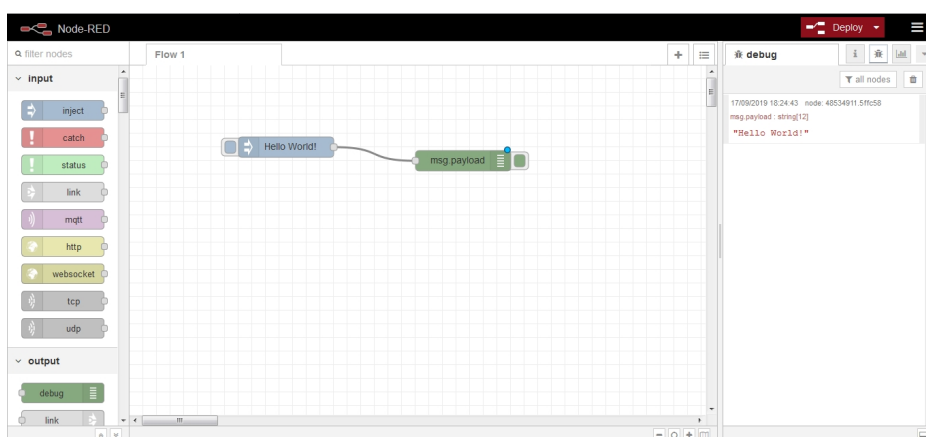
### 2.1.3.1 Node-Red

*Node-Red* trata-se de uma ferramenta de programação baseada em fluxo, usada principalmente para integrar componentes eletrônicos com aplicações de diferentes naturezas, possibilitando desenvolver programas de forma rápida e intuitiva. Ela foi originalmente criada pela equipe da *IBM's Emerging Technology Services*, fazendo parte agora do OpenJS Foundation<sup>6</sup>, que desde de setembro de 2013 passa a ter seu desenvolvimento em código aberto, realizado por meio de uma comunidade de programadores, estando assim em constante aperfeiçoamento (NODE-RED, 2019).

Essa ferramenta é executada através de um interpretador JavaScript chamado *Node.js*, no qual seu funcionamento se estabelece de forma assíncrona e orientada a eventos, permitindo rodar aplicações JavaScript do lado do servidor. A construção de programas utilizando *Node-Red* é realizada por meio de *web browser*, onde é possível acessar o editor de fluxo e o *dashboard* de visualização e operação.

A criação de uma aplicação é realizada basicamente arrastando os *nodes* (blocos que desempenham alguma função) disponíveis na paleta para a área de trabalho, onde as conexões e interações entre os diferentes *nodes* são feitas. A figura 2.4 ilustra o ambiente de programação padrão do *Node-Red*, no qual o código de fluxo é implementado por meio do botão “*Deploy*”, essa imagem ilustra um simples exemplo de envio da mensagem “*Hello World*” através de um *node* emissor de dados (*Inject*) para um *node* de exibição (*debug*), por conseguinte exibindo a mensagem na aba *debug* localizada na janela da direita.

Figura 2.4 – Ambiente *Node-Red*



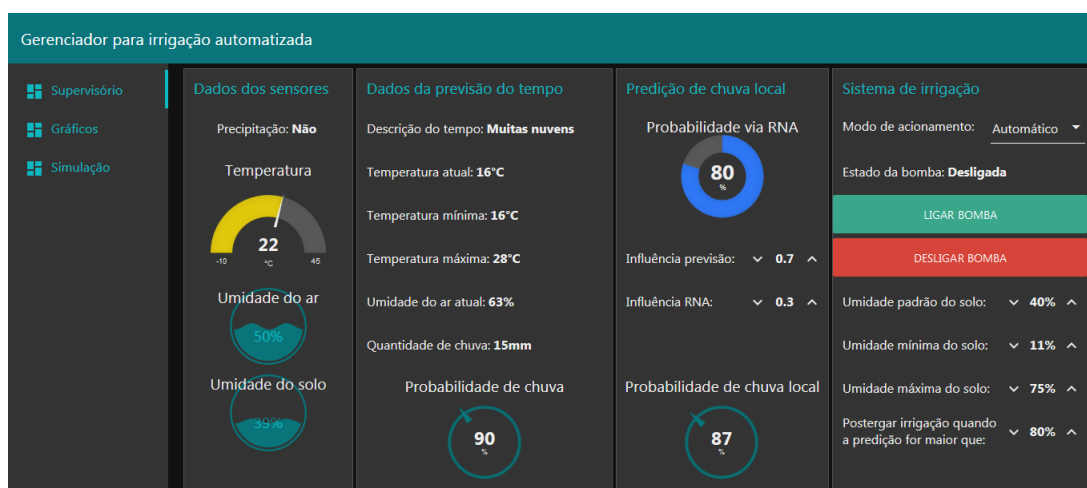
Fonte: do autor

<sup>6</sup> OpenJS Foundation: comunidade de código aberto destinada a promover a colaboração no desenvolvimento da linguagem *JavaScript* e tecnologias *web* (OPENJS, 2019).

A paleta de *nodes* padrão do *Node-Red* já fornece recursos suficientes para grande quantidade de aplicações, contudo, caso necessite de algo mais específico como: implementação de redes neurais, banco de dados SQL, processamento de imagem, dentre outros, é possível instalar mais *nodes* através da própria aplicação, que atualmente conta com mais de 2200 que podem ser incorporados a paleta padrão.

Uma forma de criar telas de supervisão e controle se estabelece por meio do conjunto de *nodes* empacotados como *node-red-dashboard*, que pode ser instalado pela própria ferramenta. Com ele é possível criar todo o *layout* do programa, atribuindo a essa tela: textos, gráficos dinâmicos, dados de sensores, campos de entrada de dados, etc. Essa é uma ótima forma de disponibilizar ao usuário um ambiente amigável de operação e monitoramento. A figura 2.5 exibe um exemplo de um *dashboard* criado a partir dessa ferramenta.

Figura 2.5 – Exemplo de *dashboard* criado por meio do *node-red-dashboard*



Fonte: do autor

Outra possibilidade que o *Node-Red* permite é a criação de seus próprios *nodes*, ou seja, com conhecimentos básicos de HTML<sup>7</sup> e *JavaScript* o usuário pode desenvolver *nodes* que desempenhem funções que atendam suas próprias necessidades, de igual forma, podendo ser compartilhado e utilizado em outros projetos, aumentando ainda mais a versatilidade da plataforma.

Devido a incorporação cada vez maior do *Node-Red* aos processos de automação, alguns dispositivos do ramo já possuem esse recurso pré-instalado, como Raspberry Pi, Predix (GE), Simatic IOT (Siemens) dentre outras plataformas que seguem o paradigma de *Internet* das coi-

<sup>7</sup> *Hypertext Markup Language (HTML)*: Linguagem de marcação destinada a ser interpretada em navegadores de internet.

sas. Com esse uso mais difundido, tende a ocorrer um crescimento da comunidade usuária e desenvolvedora, resultando assim em uma ferramenta mais completa e robusta.

### 2.1.3.2 Raspberry PI

Raspeberry Pi corresponde a computadores de dimensões próximas a de um cartão de crédito, sendo comercializado a preços reduzidos (em torno de \$30). Esse dispositivo é desenvolvido pela Raspberry Pi Foundation com o intuito de prover uma ferramenta capaz de auxiliar no aprendizado de programação e resolução de problemas através de soluções digitais e tecnológicas (RASPBERRY PI FOUNDATION, 2019).

Devida sua facilidade de uso, sua incorporação está cada vez mais presente em variadas aplicações, pois este fornece tudo o que se é esperado de um computador convencional, além de propiciar também uma interface entre o mundo físico e o digital por meio de sensores e atuadores. Atualmente existem diversos modelos e versões de Raspberry PI, permitindo assim, usos mais específicos para diferentes aplicações. O mais recente é o Raspberry Pi 4 Model B, contando com processador Broadcom BCM2711, Quad core Cortex-A72, 4 GB de memória SDRAM, rede sem fio, *Bluetooth* 5.0, conexão *Ethernet*, 4 portas USB, 40 pinos GPIO, duas saídas HDMI com suporte a 4K, etc.

Uma das grandes características dos computadores Raspberry Pi é o fato de possuírem pinos para entrada e saída de dados (GPIOs), permitindo assim sua incorporação em variadas aplicações que envolvam automatização, possuindo a função de controlador e integrador central dentro do sistema.

Para aplicações de pequeno porte, o modelo Raspeberry Pi Zero W consegue entregar bons resultados, tornando o processo de automatização simplificado e mais barato quando comparado aos controladores tradicionais. Dentre as características principais desse *hardware* estão: sistema operacional Linux; *Bluetooth* 4.1; *Wireless* 802.11 b/g/n; processador *single-core* 1 GHz; Memória RAM 512 MB; 40 pinos GPIO.

Para seu funcionamento se faz necessário um cartão micro SD com o sistema operacional Linux instalado, contudo o Raspberry já possui uma distribuição própria chamada Raspbian, compatível com todos os modelos e com algumas ferramentas já pré-instaladas que contribuem de maneira significativa no desenvolvimento das aplicações. A alimentação desse computador é feita por uma fonte de corrente contínua de 5 volts ligada através de uma porta micro USB na qual fornece também energia para todos os periféricos de entrada.

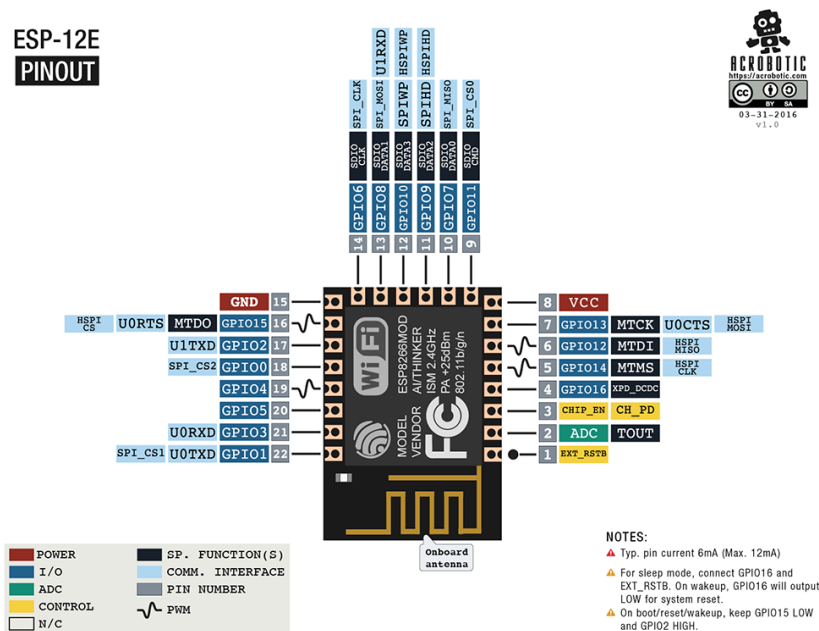


### 2.1.3.3 Chip ESP8266

O *chip* ESP8266 consiste de um dispositivo eletrônico de baixo custo com capacidade de comunicação Wi-Fi, permitindo a incorporação de diversos equipamento a rede, sendo assim, utilizado frequentemente em projetos que envolvam IOT. Atualmente existem uma variedade de módulos e placas de desenvolvimento que abrigam esse chip, tornando possível não somente a prototipação como também a sua incorporação no produto final.

Um dos módulos ESP mais utilizados atualmente é o ESP-12E, possuindo como principais características um microcontrolador de 32 bits, conversor analógico/digital de 10 bits, protocolo TCP/IP integrado, e Wi-Fi 2.4 GHz, sendo esse alimentado por 3.3 volts de tensão contínua. A pinagem referente a esse modelo é mostrado na figura 2.6, onde nota-se a presença de 16 pinos de uso geral, ou seja, podem ser configurados como entrada ou saída (GPIO), e um pino de entrada para grandezas analógicas (porta ADC).

Figura 2.6 – Módulo ESP8266-12E



Fonte: (ACROBOTIC, 2016)

Para realizar a prototipação existem algumas placas de desenvolvimento que incorporam o ESP8266, dentre as mais usadas estão a NodeMCU e a Wemos, ambas fornecem uma arquitetura que ajuda na programação e interconexão dos pinos aos dispositivos externos, sendo assim, útil durante o processo de desenvolvimento e operação.

### 2.1.3.4 Protocolo MQTT

MQTT (*Message Queuing Telemetry Transport*) compõe-se de um protocolo extremamente leve para trocas de mensagens, desenvolvido com base na pilha TCP/IP e com funcionamento seguindo o padrão *Publisher-Subscriber*. Esse foi inventado em 1999 por Andy Stanford-Clark e Arlen Nipper, sendo usado massivamente nos dias de hoje em aplicações que envolvam IOT. Por possuir grande simplicidade, esse protocolo é ideal para uso em redes de banda curta, possuindo métodos que conseguem atribuir certos graus de confiabilidade entre as trocas de mensagem, permitindo assim, a comunicação efetiva entre equipamentos dentro de uma estrutura de rede de menor robustez (MQTT.ORG, 2019).

Uma das grandes vantagens desse protocolo está no fato dele desacoplar o emissor da mensagem de seu receptor, sendo portanto, escalável mesmo em redes de menor confiabilidade. Nesse cenário existem duas entidades, o *Broker* (servidor de mensagens) e os clientes (sensores, atuadores, etc.), o primeiro tem como atribuição receber as mensagens e destiná-las aos clientes que desejam ter acesso a determinada informação, isso se estabelece através dos chamados tópicos, onde um *publisher* publica em um tópico e o *Broker* encaminha o dado a todos os *subscribers* inscritos no tópico em questão. Essas trocas de informações e comandos são estabelecidas pelos tipos de mensagens e seus parâmetros, a tabela 2.1 lista as principais.

Tabela 2.1 – Tabela dos principais comandos do protocolo MQTT

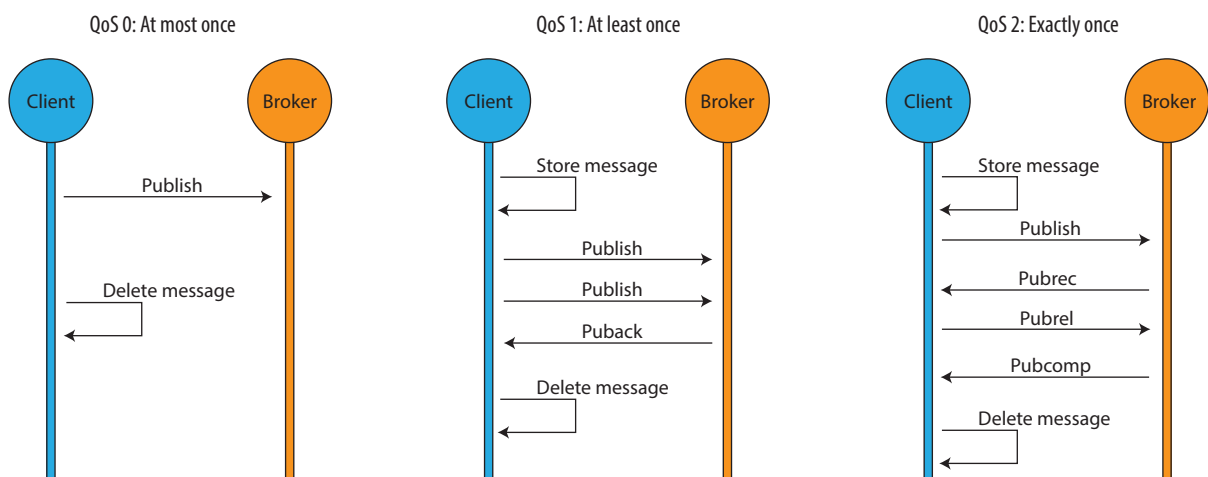
Mensagem	Descrição	Parâmetros
CONNECT	Usada para estabelecer a conexão entre o <i>broker</i> e o cliente	packedId, cleanSession, username, password, lastWillTopic, lastWillQos, lastWillMessage, keepAlive.
CONNACK	Resposta do cliente ao <i>broker</i> para estabelecimento da conexão	packedId, sessionPresent, returnCode.
SUBSCRIBE	Mensagem do cliente ao <i>broker</i> indicando o tópico de inscrição	packedId, qos, topic.
SUBACK	Resposta do <i>broker</i> ao cliente referente a sua inscrição no tópico	packedId, returnCode.
UNSUBSCRIBE	Usada para retirar sua inscrição de um tópico	packedId, topic.
UNSUBACK	Resposta do <i>broker</i> ao cliente referente a remoção de sua inscrição do tópico.	packedId.
PUBLISH	Mensagem do cliente ao <i>broker</i>	packedId, retainFlag, topicName, qos, payload.

Um ponto importante a respeito da comunicação realizada entre *Broker* e clientes consiste no chamado *Quality of Service* (QoS), esse define a garantia da entrega das mensagens, possuindo no protocolo MQTT três níveis, sendo eles:

- QoS 0: método mais rápido, porém o menos confiável. Quando configurado dessa forma a mensagem é enviada uma única vez e sem confirmação de recebimento;
- QoS 1: nessa configuração a mensagem é garantida de ser entregue pelo menos uma vez, parando o envio quando a confirmação de recebimento for identificada;
- QoS 2: esse método garante que a mensagem seja entregue apenas uma vez, sendo portanto o método mais lento já que mais mensagens de confirmação são necessárias.

A Figura 2.7 mostra a estrutura das trocas de mensagens para os diferentes níveis de QoS usados no protocolo MQTT. Nela é possível observar a comunicação entre o *publisher* e o *Broker*, contudo a mesma configuração de mensagens também é verificada entre o *Broker* e o *subscriber*, uma vez que a configuração de QoS pode ser feita de forma independente entre *publisher-Broker* e *Broker-subscriber*.

Figura 2.7 – Níveis de QoS usados no protocolo MQTT



Fonte: do autor

Para que a comunicação aconteça no protocolo MQTT a presença de um *Broker* é indispensável, nesse trabalho foi utilizado o Mosquitto<sup>8</sup> para o cumprimento dessa tarefa.

<sup>8</sup> Mosquitto: *broker* de código aberto, pertencente a Eclipse Foundation. Possuindo suporte as versões 5.0, 3.1.1 e 3.1 do protocolo MQTT.

## 2.2 Metodologia

Com base nos estudos das tecnologias e ferramentas associadas a quarta revolução industrial, foi observado alguns ambientes nos quais a implementação do *Node-Red* se estabelece de forma adequada. A seguir serão expostas as características presentes nesse *software* que auxiliam no processo de automatização, evidenciando alguns pontos-chaves dessa ferramenta no cenário tecnológico atual. Por fim, foram simulados três ambientes em diferentes setores em que o *Node-Red* pôde ser aplicado. Nesses projetos serão apresentados os fluxos usados, bem como, as simulações e resultados obtidos.

### 2.2.1 Importância do *Node-Red* no cenário atual

Baseado na grande quantidade de aplicações das tecnologias ligadas a quarta revolução industrial, é possível observar que em um único ambiente pode existir a sobreposição de várias dessas tecnologias, ou seja, em uma aplicação agrícola, pode estar presente o uso de robótica, inteligência artificial, IOT, dentre outros; tudo isso em uma aplicação. É nesse tipo de cenário que a ferramenta *Node-Red* mostra sua importância.

*Node-Red* possui a capacidade de fornecer em um único ambiente uma grande quantidade de recursos. Abaixo são listados alguns desses:

- Interação com a *internet*: presença de *nodes* que possibilitam a obtenção de dados da *internet* de forma simplificada, seja por requisições HTTP, *webService*, *WebSocket*, TCP, UDP, etc. Possuindo até recursos de interação com redes sociais e *e-mails*;
- Comunicação: uma vez que essa ferramenta é orientada a IOT, o protocolo de comunicação MQTT já se encontra presente, possuindo também recursos de obtenção de dados através de portas físicas do *hardware* no qual este está sendo executado, como por exemplo, quando instalado em Raspberry permite a leitura e escrita direta em seus pinos GPIO;
- Armazenamento de dados: possui funções de fácil uso de escrita e leitura de arquivos, podendo assim, armazenar informações e configurações durante a operação, de forma local ou até mesmo interagir com algum banco de dados *on-line*, como o *mySQL*;
- Tratamento dos dados: *Node-Red* possui *nodes* que facilitam o tratamento dos dados durante a operação, aplicando assim, filtros e molduras que auxiliam na obtenção da informação;

- Execução de comandos externos: recurso que permite executar ações dentro do sistema operacional, como abrir programas, rodar processos externos, ou seja, interagir com alguma outra ferramenta presente no ambiente de execução;
- Criação de código: além de todos os *nodes* com funções já determinadas na ferramenta, é possível criar seus próprios códigos por meio de um *node* chamado *function*, permitindo adicionar código em *JavaScript* personalizado a sua aplicação, dessa forma, não limitando o usuário ao uso dos recursos já pré-existentes.

Os recursos citados correspondem a algumas das funções nativas presentes na plataforma, contudo, ainda é possível encontrar e instalar outras ferramentas que estão em constante desenvolvimento e que possam contribuir em aplicações mais específicas, na comunidade desenvolvedora existem recursos como: redes neurais artificiais, visão computacional, reconhecimento de voz, ferramentas de criação de interfaces gráficas, etc.

Apenas com os recursos nativos do *Node-Red* já se faz executável diversos programas, contudo a opção de adicionar mais funcionalidades ou desenvolver seus próprios códigos, faz dessa ferramenta uma plataforma expansível de acordo com a necessidade de cada aplicação, ideal para o cenário de grande dinamismo da quarta revolução industrial.

Um grande benefício do *Node-Red* está no fato de este ser de código aberto, permitindo não apenas sua modificação, como também a inclusão de recursos próprios e personalizados, conseqüentemente, não necessitando de outro *software* caso o processo no qual está sendo aplicado venha a se alterar.

Por se tratar de uma plataforma de desenvolvimento por meio da criação de fluxos, *Node-Red* possui a vantagem de ser de fácil aprendizado e entendimento, uma vez que o caminho da informação é exposto ao usuário mais claramente, possibilitando também modificações sem a necessidade de grandes conhecimentos quando comparado a outras ferramentas de programação tradicionais.

Dado que as tecnologias ligadas a quarta revolução industrial se fazem presentes em diversas áreas do conhecimento, elas atingem pessoas que por ventura não possuem domínio de programação tradicional suficiente para desenvolver aplicações que permitam a interação dessas tecnologias, dessa forma, *Node-Red* reduz a distância entre desenvolvedor e consumidor final, permitindo que o próprio usuário da ferramenta possa compreendê-la e modificá-la de acordo com suas necessidades, sem obrigatoriamente depender de empresas externas para realizar tal tarefa.

## **2.2.2 Exemplo de aplicação do *Node-Red* no setor agrícola**

O setor agrícola no Brasil embora ainda apresente baixo nível de automação, várias soluções e dispositivos tecnológicos estão sendo desenvolvidos para essa área. O exemplo a seguir consiste da aplicação de *Node-Red* versão 0.20.7 para auxiliar no processo de irrigação automática.

### **2.2.2.1 Projeto**

O objetivo desse projeto é fornecer ao usuário um programa capaz de realizar o processo de irrigação de forma automática e inteligente, na qual seja possível também, visualizar e alterar parâmetros do sistema. Como parâmetros principais tem-se: umidade do solo, temperatura do ar, previsão do tempo, umidade do ar e presença de chuva. Com base nesses dados o sistema escolhe a melhor forma de realizar a irrigação, a fim de economizar água, energia e tempo de máquina, tornando assim o processo mais eficiente.

Esse projeto lida com diferentes tecnologias, dentre as principais se destacam: o uso do protocolo MQTT para a comunicação dos sensores e atuadores com o *Broker*; requisição HTTP a fim de obter dados de previsão do tempo *on-line* e redes neurais artificiais (não implementado, apenas simulado) para auxiliar na tomada de decisão.

Esse sistema conta com dois ambientes, um de operação e outro de desenvolvimento, sendo o primeiro referente a tela na qual o usuário pode visualizar os dados dos sensores, modificar os parâmetros do sistema, acionar atuadores, verificar condições do clima, dentre outros; já o ambiente de desenvolvimento é destinado a realizar mudanças estruturais no programa, é nesse espaço que ocorre a programação e toda a lógica de seu funcionamento, de igual forma, sendo também responsável pela criação da tela de operação.

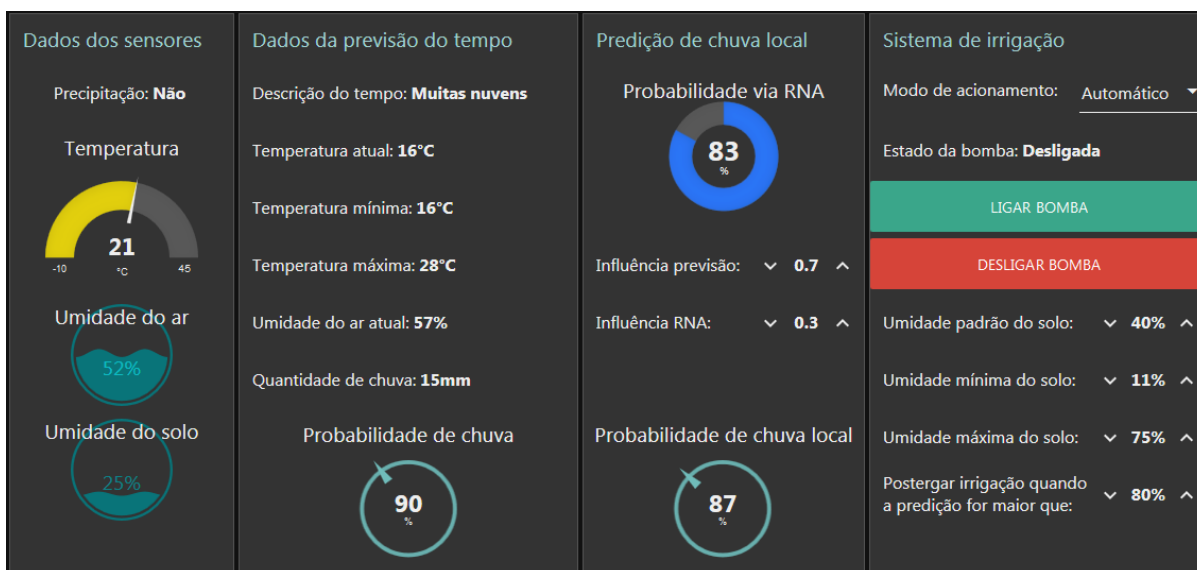
### **2.2.2.2 Ambiente de operação**

O ambiente de operação desse projeto possui duas telas, uma de supervisão e controle, e outra de gráficos, na qual é mostrado os dados dos sensores ao longo do tempo.

## Tela de supervisão e controle

A figura 2.8 ilustra a tela de supervisão e controle, na qual observa-se quatro painéis, sendo eles: dados dos sensores, dados da previsão do tempo, previsão de chuva local e sistema de irrigação. A seguir é descrito a funcionalidade de cada um.

Figura 2.8 – Tela de supervisão e controle



Fonte: do autor

- Dados dos sensores: exibe informações fornecidas pelos sensores instalados no campo, como: precipitação, temperatura do ar, umidade do ar e umidade do solo;
- Dados da previsão do tempo: mostra informações provindas da *internet* a respeito das previsões climáticas da região em questão;
- Predição de chuva local: por meio de dados coletados durante a operação, é possível o treinamento de uma rede neural artificial (RNA) ao longo do tempo, relacionando dados da previsão do tempo e dos sensores, com o intuito de fornecer uma estimativa mais apurada da probabilidade de chuva local, nesse painel também é possível determinar a influência de cada probabilidade (previsão via *internet* e via RNA) a ser considerada;
- Sistema de irrigação: nesse painel é configurado o modo de acionamento, os comandos de ligar e desligar a bomba, estado da bomba e os parâmetros de acionamento.

## Tela de gráficos

Nesse ambiente é possível visualizar de maneira simplificada como a temperatura, presença de chuva, umidade do ar e do solo, variaram ao longo do tempo. A figura 2.9 exibe uma simulação de variações nos valores lidos pelos sensores ao longo do tempo.

Figura 2.9 – Tela de gráficos



Fonte: do autor

### 2.2.2.3 Ambiente de desenvolvimento

O ambiente de desenvolvimento do programa foi dividido em seis abas, sendo elas: *Main*, *Dashboard*, *Charts*, *Initial state*, *Database* e *Simulation*. Essa divisão embora não necessária, auxilia na organização dos elementos do sistema, tornando mais fácil a modificação futura. A seguir serão explicadas as etapas de funcionamento, bem como, os principais *nodes* e funções utilizadas.

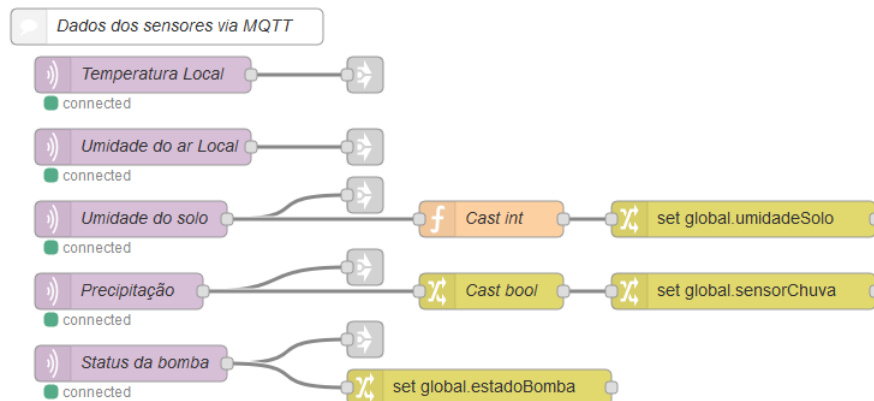
### Aquisição dos dados dos sensores

A figura 2.10 ilustra a parte do programa responsável por receber as informações provenientes dos sensores via MQTT. Nesse fragmento a informação dos sensores são conectadas a um *link*, que serve apenas de ponte para outra aba do programa. Uma vez que as informações



provindas via MQTT são do tipo *String*(não numérico), em situações nas quais operações entre os dados foram necessárias adicionou-se um bloco para transformar o tipo da informação (*cast*). Outra operação evidenciada nessa parte, corresponde a atribuição de alguns dados em variáveis globais, para que esses sejam acessíveis em todas as partes do programa.

Figura 2.10 – Dados dos sensores via MQTT

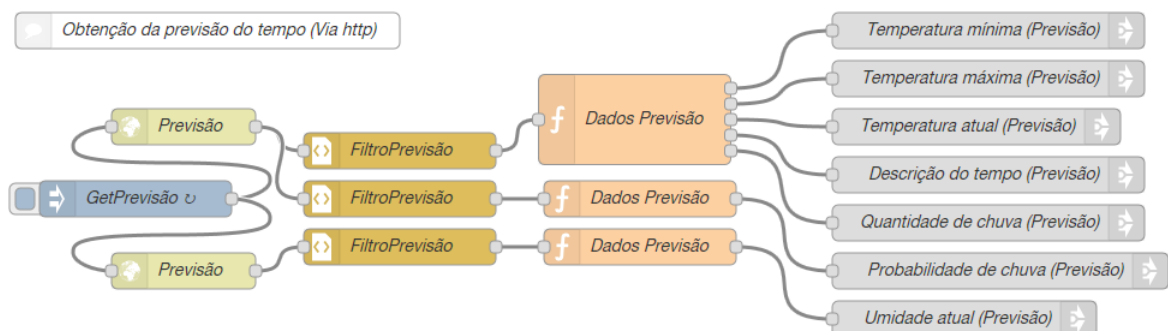


Fonte: do autor

### Aquisição dos dados da previsão do tempo

A previsão do tempo foi obtida através do *site www.climatempo.com.br*, no qual por meio de uma simples requisição HTTP e implementação de filtros, foi possível adquirir os dados de previsão para a cidade configurada, podendo esses dados serem atualizados na frequência desejada. As informações obtidas foram: temperatura atual, temperatura mínima e máxima para o dia, probabilidade de chuva, quantidade de chuva, umidade do ar e descrição do tempo. A figura 2.11 ilustra os elementos dessa parte.

Figura 2.11 – Dados da previsão do tempo



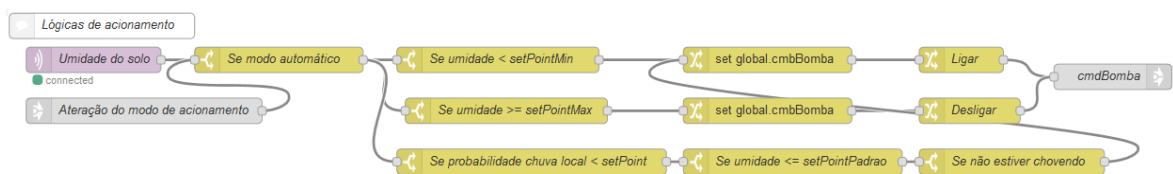
Fonte: do autor

Dentre os *nodes*, observa-se a presença de um comando cíclico representado pelo bloco “GetPrevisão”, no qual é solicitado aos blocos “Previsão” acesso aos *sites* configurados, após a obtenção dos dados do *site*, a informação é passada pelos filtros e pelas funções que isolam os dados de interesse, para então serem usados em outras partes do programa.

### Lógica de acionamento do sistema de irrigação

Essa parte corresponde a lógica na qual o acionamento do sistema de irrigação está condicionado. A figura 2.12 exibe quais são os critérios de acionamento e como a informação caminha pelos blocos do tipo *switch*, até chegar no *link* de comando do sistema (fluxo pertencente a aba *Main*).

Figura 2.12 – Lógica de acionamento do sistema de irrigação



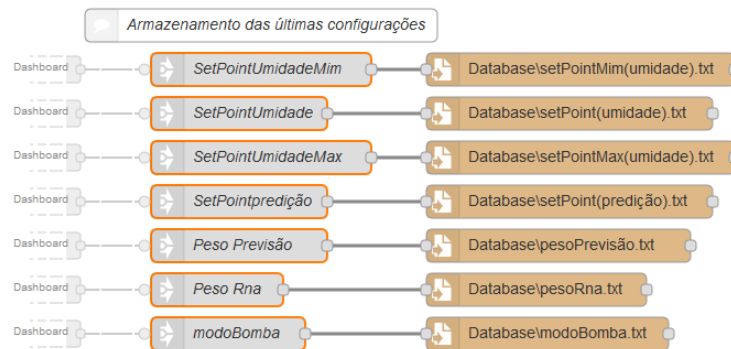
Fonte: do autor

De forma simplificada, o sistema é acionado em duas condições: a primeira diz respeito ao momento em que o solo atinge a umidade padrão configurada e ao mesmo tempo a previsão mostra probabilidade de chuva abaixo da escolhida e não se verifica presença de chuva pelo sensor; na segunda condição o acionamento é feito quando a umidade alcança o valor mínimo estabelecido. Esses dois critérios de acionamento só são válidos quando o sistema estiver em modo automático.

### Armazenamento das configurações do sistema

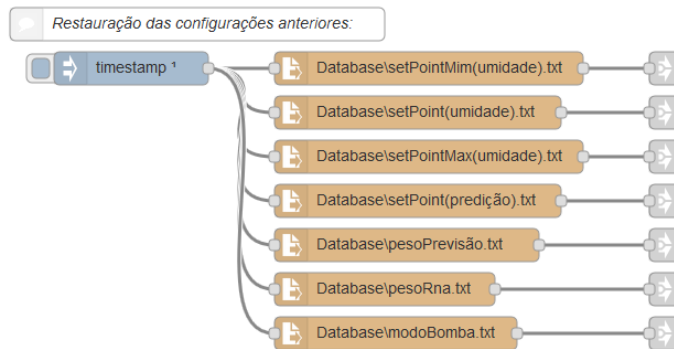
Para que o programa consiga restaurar as configurações escolhidas na tela de supervisão e controle (influência da previsão e RNA, umidade padrão, postergar irrigação, etc.), foi utilizado a abordagem de gravação desses dados em arquivos de texto a cada alteração realizada pelo usuário, sendo então, carregadas automaticamente no início da execução do programa. As figuras 2.13 e 2.14 ilustram os fluxos correspondentes, no qual o primeiro está presente na aba “Database” e o segundo na aba “InicialState”.

Figura 2.13 – Gravação das configurações



Fonte: do autor

Figura 2.14 – Carregamento das configurações

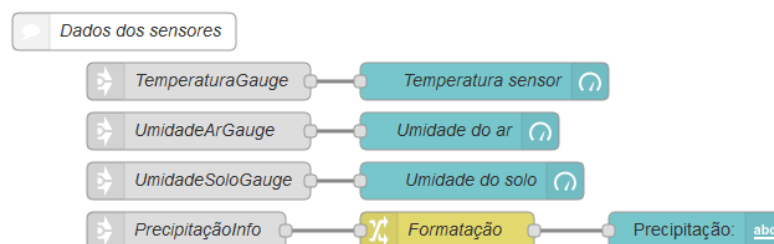


Fonte: do autor

## Fluxo da interface de supervisão e controle

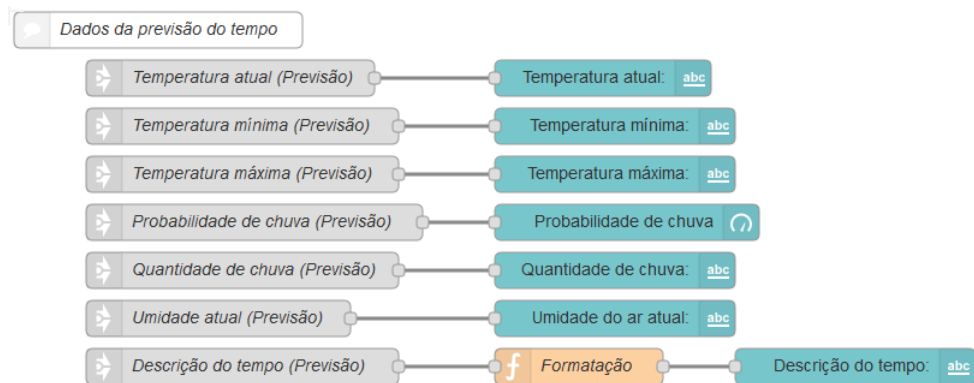
Nessa parte foi adicionado cada elemento presente no ambiente de supervisão, como: gráficos, controles, informações dos sensores, campo de entrada de dados, botões, dentre outros. As figuras 2.15 e 2.16 ilustram como é simples a estrutura de blocos necessária para exibir valores, textos e gráficos em uma interface gráfica.

Figura 2.15 – Fluxo referente a exibição das informações dos sensores



Fonte: do autor

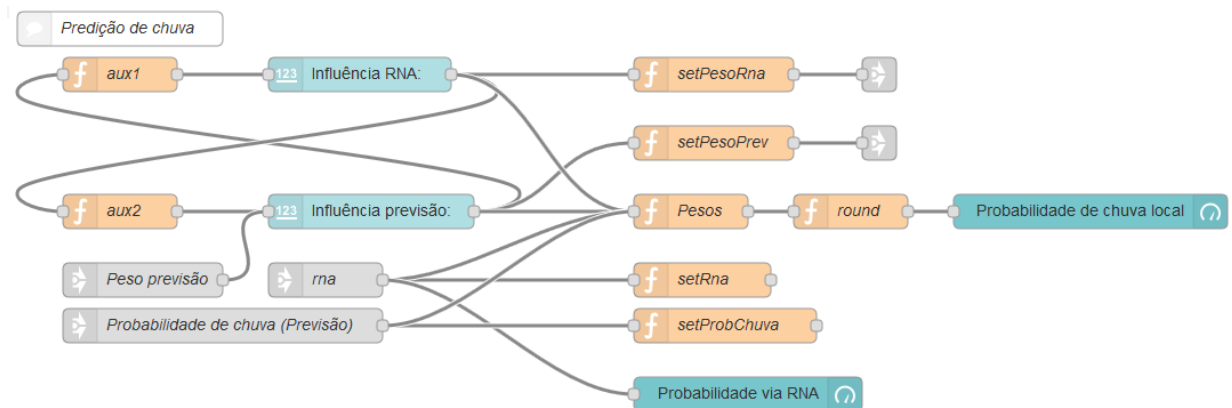
Figura 2.16 – Campos de textos e gráficos referentes aos dados da previsão do tempo



Fonte: do autor

A figura 2.17 ilustra a organização dos *nodes* que foi necessária para criar os controles e os gráficos referente ao painel de previsão de chuva, apresentando blocos que carregam as informações anteriores, bem como funções auxiliares para permitir alterações lógicas entre os pesos da previsão via *internet* e RNA.

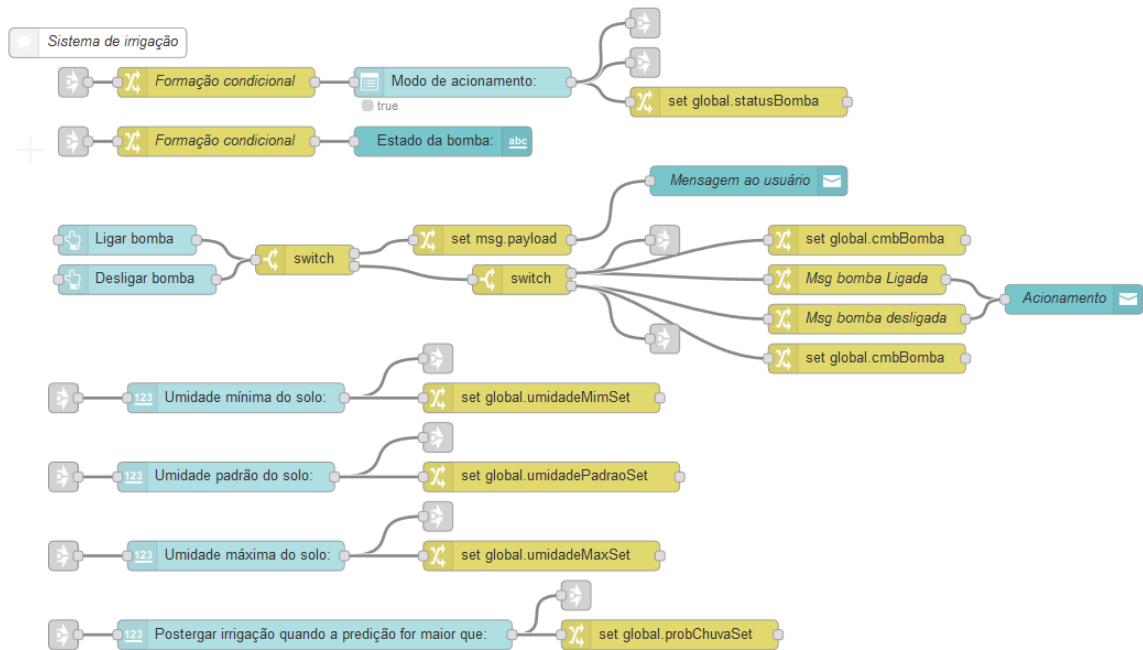
Figura 2.17 – Fluxo referente ao painel previsão de chuva



Fonte: do autor

Para o painel “Sistema de irrigação” foi utilizado o fluxo ilustrado na figura 2.18, apresentando blocos de informações de: modo de acionamento, lógicas de travamento de operação da bomba quando o sistema estiver em modo automático, botões de comando e campos de entrada de dados dos *set points*.

Figura 2.18 – Fluxo referente ao painel Sistema de irrigação

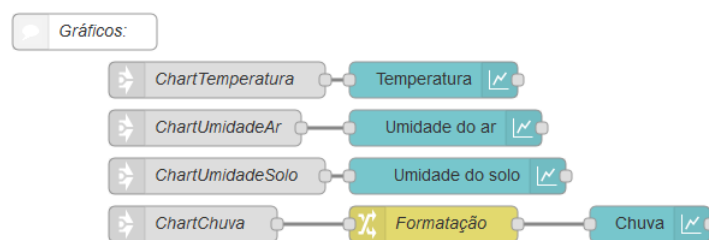


Fonte: do autor

### Fluxo da tela de gráficos

A tela de gráficos se apresenta de forma bem simplificada, possuindo fluxo ilustrado na figura 2.19, na qual é possível observar *inputs* provindos dos sensores, demonstrando também um bloco de formatação (utilizado para transformar informação do tipo *true/false* para 1 ou 0).

Figura 2.19 – Fluxo referente a representação gráfica dos dados dos sensores



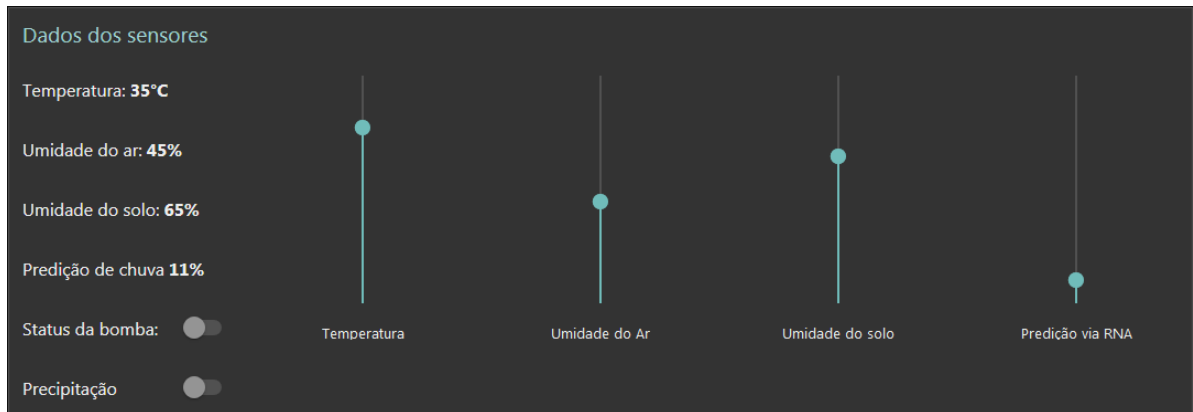
Fonte: do autor

#### 2.2.2.4 Resultados e simulação

Esse projeto foi simulado por meio de uma tela de simulação (figura 2.20), onde foi possível simular dados provenientes de sensores, enviando as informações via protocolo MQTT, inserindo assim, dados de temperatura, umidade do ar, umidade do solo, previsão via RNA,

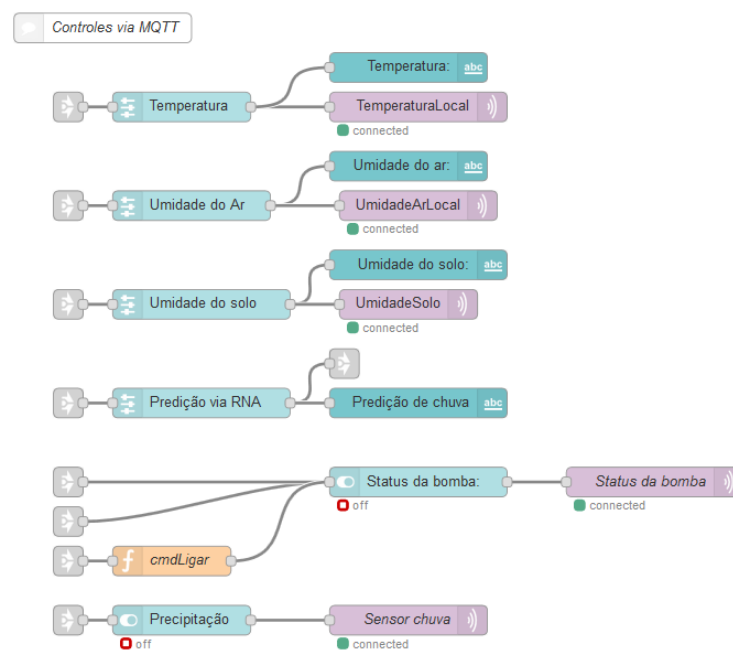
*status* da bomba e presença de chuva. Desse modo, foi verificado e validado as lógicas configuradas no programa. O fluxo executado pela tela de simulação é ilustrado na figura 2.21.

Figura 2.20 – Tela de simulação



Fonte: do autor

Figura 2.21 – Fluxo pertencente a tela de simulação



Fonte: do autor

Por meio da simulação foi verificado um comportamento que era esperado do sistema, o qual obedeceu a todas as regras estabelecidas pelos fluxos criados, acionando o sistema de irrigação de forma adequada aos valores dos sensores, da previsão do tempo e dos parâmetros configurados.

### 2.2.3 Exemplo de aplicação do *Node-Red* no setor industrial

O setor industrial é uma das áreas mais afetadas pelas novas tecnologias da Indústria 4.0, tornando esse ambiente um local de alta conectividade e interação entre os dispositivos do chão de fábrica e os *softwares* de gestão. O exemplo a seguir consiste da aplicação do *Node-Red* versão 0.20.7 no controle da qualidade de produtos em uma esteira. Embora esse cenário industrial seja clássico, foi possível observar como o *Node-Red* consegue atuar em situações já tradicionais, permitindo a fácil integração com outros elementos que possam vir a ser incorporados ao processo.

#### 2.2.3.1 Projeto

O objetivo desse projeto consiste em fornecer um sistema capaz de realizar o controle da qualidade de produtos por meio de seu peso, separando assim os itens em duas categorias (produtos *OK* e *NOK*), permitindo assim, que os que estiverem dentro da faixa de peso aceitável possam continuar no processo.

O sistema conta com uma esteira para transporte dos produtos, uma balança incorporada ao caminho da esteira, um soprador para alterar a rota dos produtos que estiverem fora de especificação e sensores de detecção do produto. Nesse exemplo todos os dispositivos se comunicam via protocolo MQTT, contudo, também é possível a comunicação física entre atuador e controlador para situações mais específicas.

Nesse programa é fornecida uma interface para definir qual o peso característico do produto e sua variação permitida, sendo assim, quando o produto não estiver dentro dessa faixa um sinal é enviado ao soprador para que esse atue, retirando assim o produto da linha de produção principal. Também é possível definir um limite máximo de produtos não conformes, que em caso de este ser alcançado, uma notificação será mostrada ao usuário e a esteira será desligada, retornando seu processo quando o operador ligar o sistema novamente.

Outro recurso fornecido por esse programa é o de gerenciamento da manutenção dos equipamentos, possuindo mecanismo de acionamento automático da empresa responsável pela manutenção do equipamento, informando a ela a atual condição do dispositivo.

Esse programa conta com dois ambientes, um de operação e outro de desenvolvimento. O ambiente de operação fornece toda a informação necessária para o controle e gestão da produção, ao passo que o de desenvolvimento é o local no qual a lógica do programa foi construída.

A seguir são mostradas as telas do ambiente de operação, bem como, os *nodes* de programação presentes no ambiente de desenvolvimento.

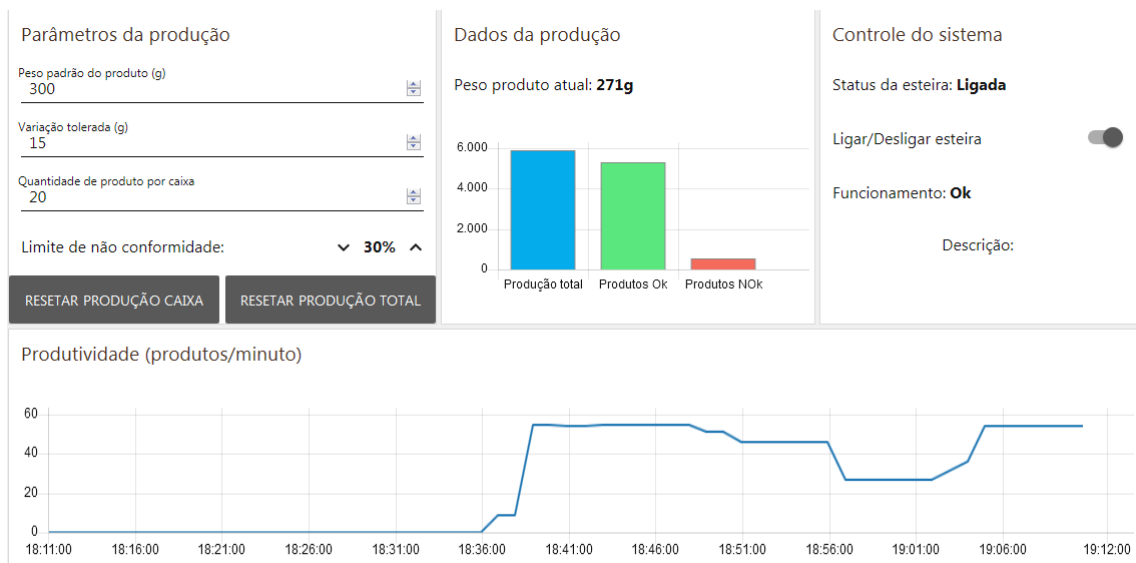
### 2.2.3.2 Ambiente de operação

Esse ambiente possui duas telas, sendo elas: tela de produção e tela de equipamentos. A primeira refere-se ao monitoramento e controle da linha de produção, já a segunda é responsável pela gestão dos equipamentos presentes na esteira.

#### Tela de supervisão e controle

A figura 2.22 exibe a tela de supervisão e controle, na qual observa-se quatro painéis, sendo eles: parâmetros da produção, dados da produção, controle do sistema e produtividade. A seguir é descrito a funcionalidade de cada um.

Figura 2.22 – Tela de supervisão e controle



Fonte: do autor

- Parâmetros da produção: possui campos de entrada de dados para definir as características de desejo do produto, sendo elas: peso do produto, variação tolerada, quantidade de item por caixa e limite de não conformidade;
- Dados da produção: exibe informações da produção (produção total, produtos *OK* e produtos *NOK*) em gráfico de barras, mostrando também o peso do produto lido pela balança;



- Controle do sistema: painel responsável por fornecer o controle da esteira e exibir informações sobre *status* de funcionamento do sistema;
- Produtividade: exibe informação visual da produtividade ao longo do tempo. Esse tipo de visualização possui grande utilidade para identificar períodos de baixa produtividade, servindo como ponto inicial na busca do problema.

### Tela de gestão da manutenção dos equipamentos

Nessa tela (figura 2.23) se faz possível visualizar e configurar o tempo de manutenção dos equipamentos, determinando os parâmetros para a manutenção. Com base nessas informações é feito o envio automático de *e-mail* para o setor ou empresa responsável pela manutenção do equipamento em questão.

Figura 2.23 – Tela de gestão da manutenção dos equipamentos



Fonte: do autor

### 2.2.3.3 Ambiente de desenvolvimento

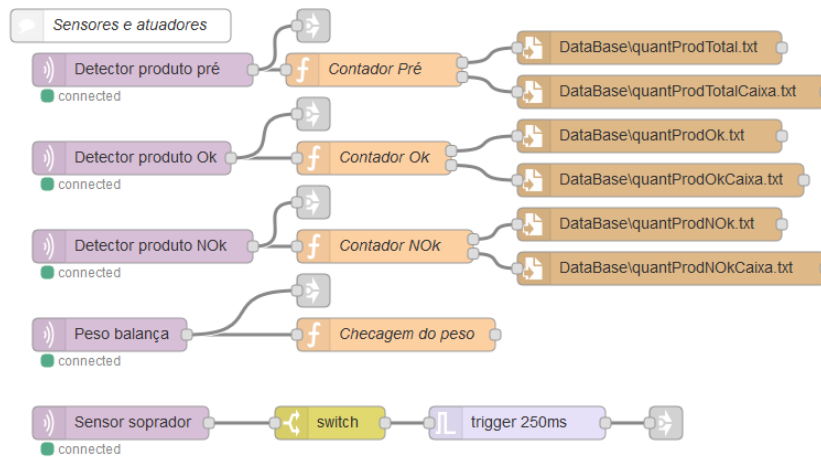
A seguir serão descritos os fluxos que foram utilizados para a construção do programa.

#### Aquisição dos dados dos sensores

Parte do programa responsável por receber a leitura dos sensores presentes na esteira, como: detectores de presença e medidores de peso. A figura 2.24 ilustra o fluxo encarregado por essa função do programa. Nele observa-se que cada sinal de detecção é incrementado a

contagem da produção (funções), de modo que as informações não se percam ao reiniciar o sistema, esses dados são então gravados em arquivos de texto, sendo assim, carregados durante o início do programa.

Figura 2.24 – Fluxo responsável pela aquisição dos dados dos sensores

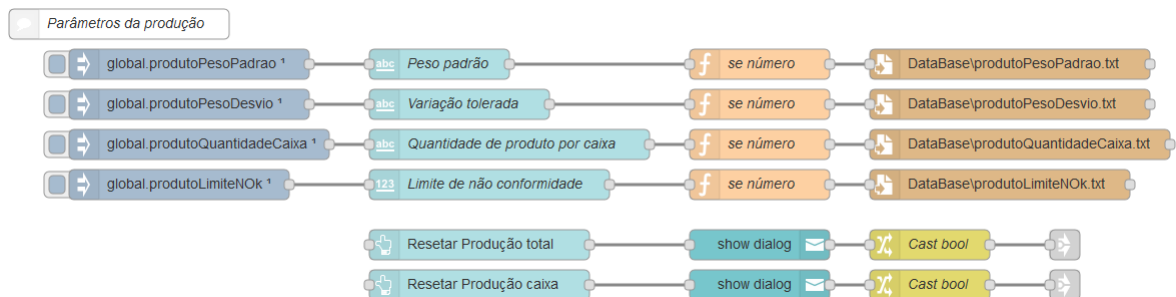


Fonte: do autor

## Parâmetros da produção

O fluxo mostrado na figura 2.25 é responsável por receber os dados de entrada do usuário referente aos parâmetros da produção, realizando checagens e armazenamento das informações. Também é possível visualizar os blocos correspondentes aos botões para zerar os dados de produção, sendo esses ligados a blocos de mensagem de confirmação da ação.

Figura 2.25 – Fluxo responsável pelo painel de parâmetros da produção

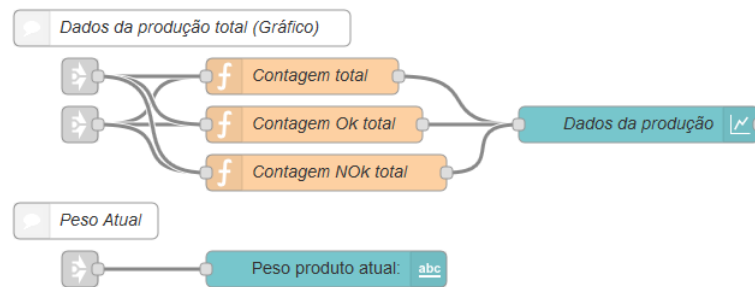


Fonte: do autor

## Dados da produção

Fluxo responsável por exibir o peso do produto atual e os dados da produção de forma gráfica, recebendo como *inputs* informações de contagem global adquiridas pelas funções, sendo essas atualizadas a cada evento dos sensores. A figura 2.26 ilustra o trecho citado.

Figura 2.26 – Fluxo responsável pelo painel de dados da produção

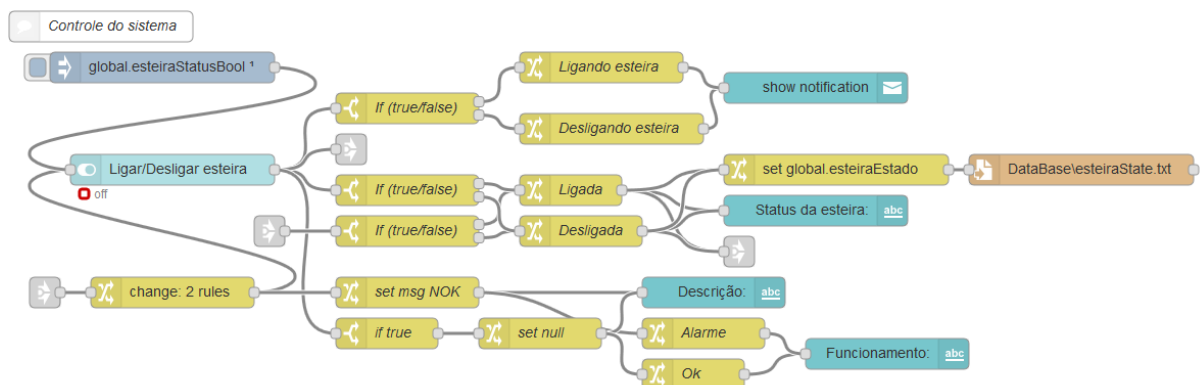


Fonte: do autor

## Controle da esteira

Parte do programa responsável pela lógica de acionamento, alarmes e notificações a respeito do funcionamento do sistema. Esse fluxo é ilustrado na figura 2.27, onde observa-se blocos de condições para exibir notificações específicas para cada situação, nota-se também *nodes* de definição de variável global e armazenamento de estado.

Figura 2.27 – Fluxo responsável pelo painel de controle da esteira

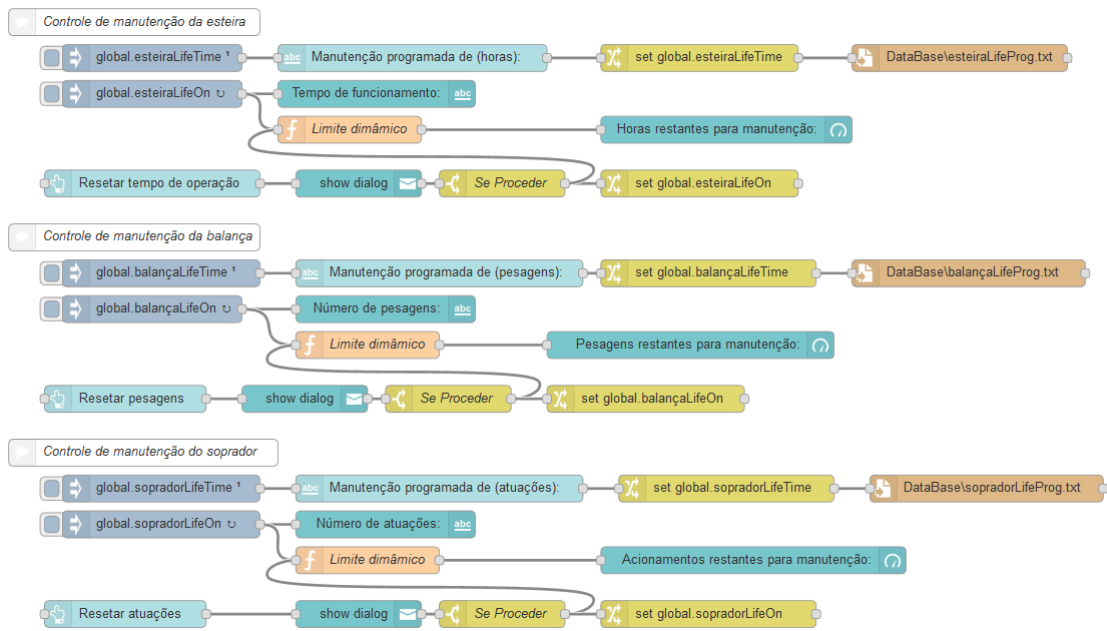


Fonte: do autor

## Manutenção dos equipamentos

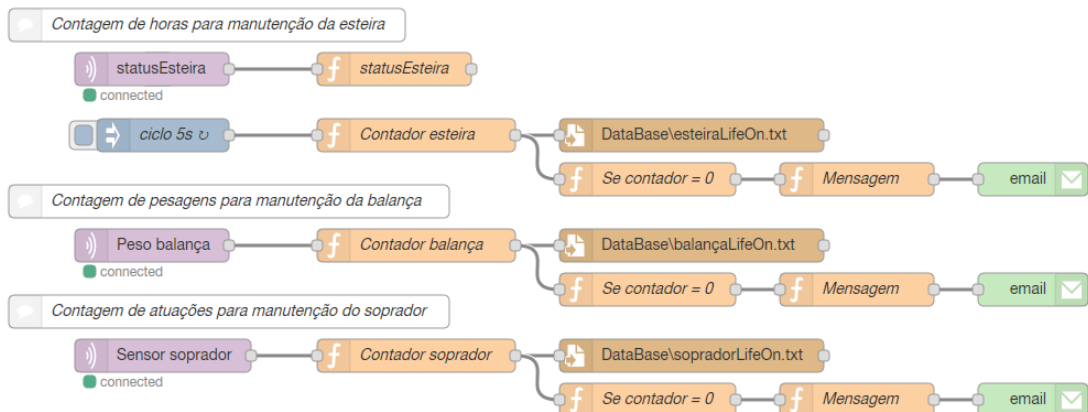
As figuras 2.28 e 2.29 ilustram os fluxos responsáveis por exibir e controlar a tela de equipamentos. Dentre os principais blocos utilizados estão: entrada de dados, medidores gráficos, gravadores em disco e mensagens de confirmação. Esses blocos trabalham de forma integrada, a fim de fornecer uma interface dinâmica e intuitiva ao usuário.

Figura 2.28 – Fluxo responsável pela tela de manutenção dos equipamentos



Fonte: do autor

Figura 2.29 – Fluxo responsável pela lógica e gestão da manutenção dos equipamentos

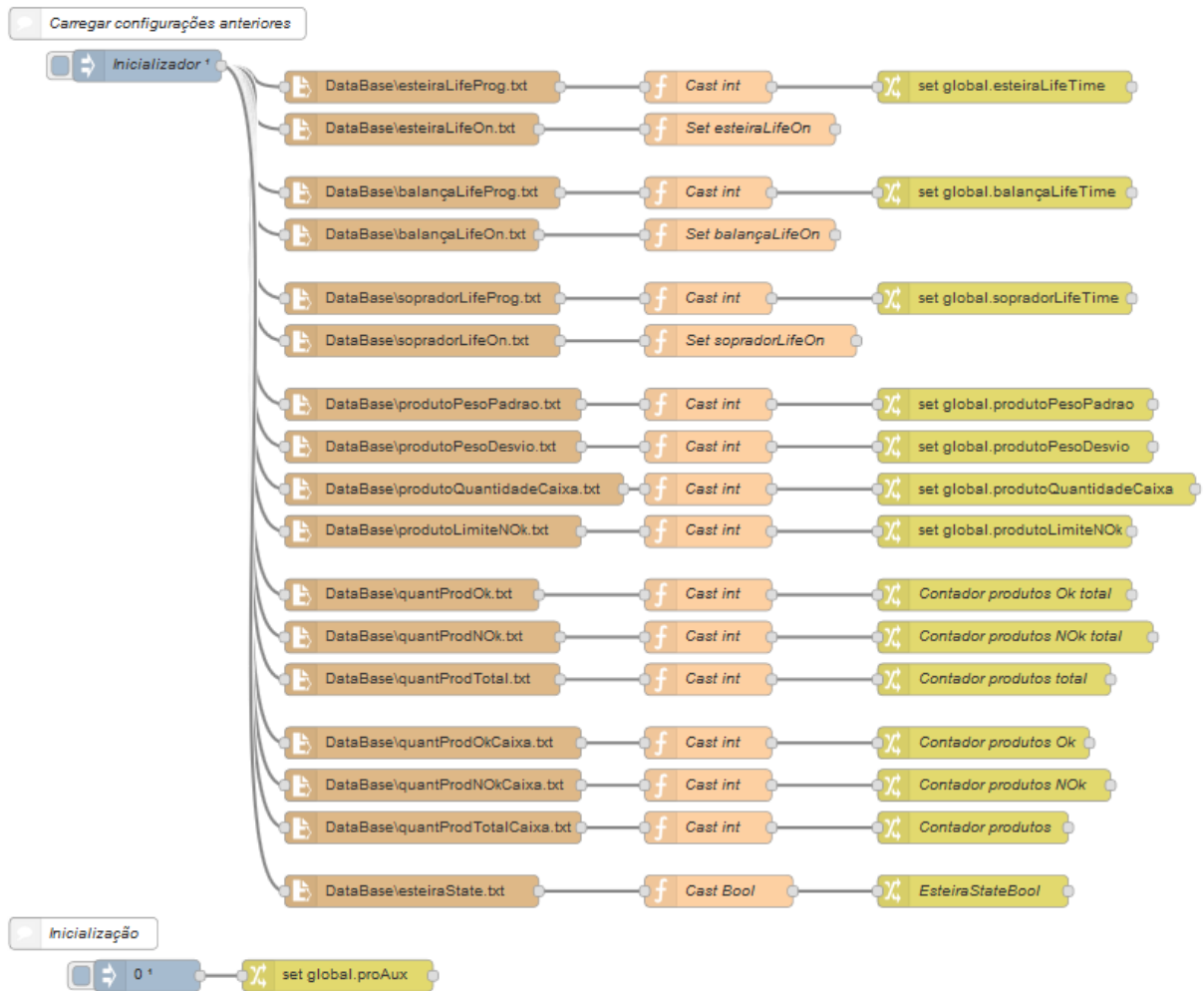


Fonte: do autor

## Inicializações

De modo a fazer com que o programa restaure as configurações previamente realizadas, se fez necessário o fluxo ilustrado na figura 2.30, no qual é observado blocos para carregar arquivos, funções para converter o tipo do dado e por fim, blocos de atribuição de valores à variáveis globais.

Figura 2.30 – Fluxo responsável pelas inicializações do sistema

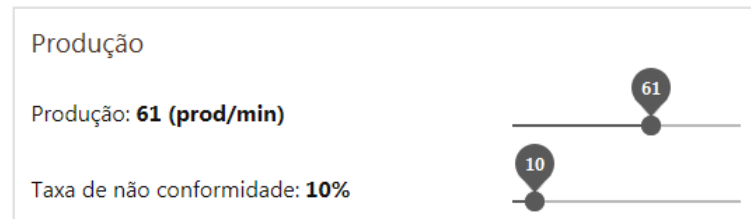


Fonte: do autor

### 2.2.3.4 Resultados e simulação

Esse projeto foi simulado por meio de uma tela (figura 2.31) na qual é possível determinar a velocidade de produção e a taxa de não conformidade, verificando assim a resposta do programa a esses eventos.

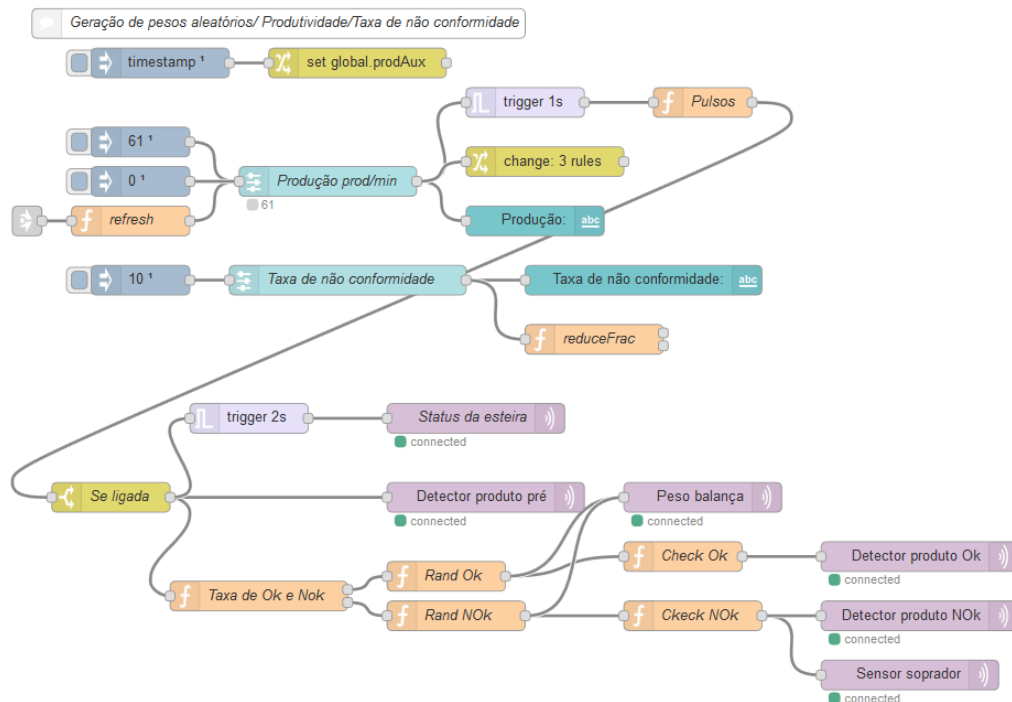
Figura 2.31 – Tela de simulação da produção



Fonte: do autor

O fluxo responsável pela simulação é mostrado na figura 2.32. A seguir são descritos os principais blocos pertencentes a este.

Figura 2.32 – Fluxo responsável pela simulação da produção



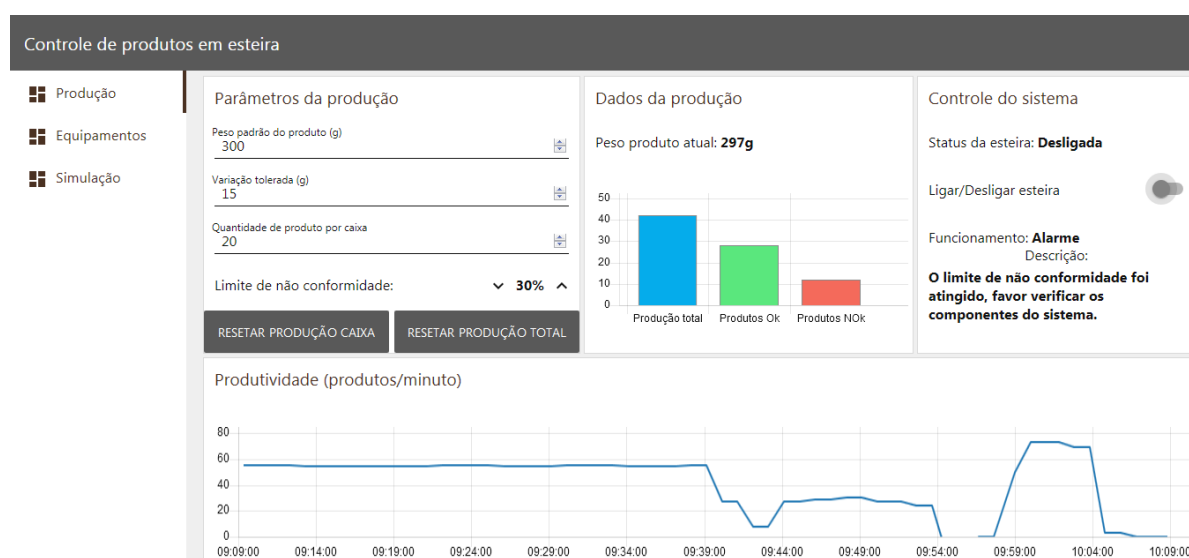
Fonte: do autor

- Inicializadores: atribuem valores predeterminados de produção ao iniciar o programa. Exemplo: produção por minuto igual a 61, taxa de não conformidade igual a 10;
- Gerador de pulsos: a função “Pulso” tem como finalidade simular a entrada de produto na esteira, sendo a frequência desse determinada pela taxa de produção;
- Gerador de pesos: as funções “Taxa de Ok e Nok”, “Rand Ok” e “Rand NOK”, trabalham de modo a gerar dados de peso aleatórios dentro do especificado no bloco de não conformidade;

- Checagem do peso: as funções “*Check Ok*” e “*Check Nok*” tem como atribuição verificar se o peso está dentro ou fora do especificado pelo usuário, simulando assim a detecção de um produto com característica aceitável ou não aceitável.

Por meio de simulações foi verificado um comportamento adequado do programa, tanto na parte de exibição dos dados quanto na parte funcional do sistema. A figura 2.33 ilustra a tela de operação e controle quando a taxa máxima de não conformidade é atingida. Por meio dela, verifica-se que a esteira é desligada automaticamente e uma mensagem é mostrada ao usuário, informando a situação atual e a descrição do que ocorreu.

Figura 2.33 – Simulação de taxa máxima de não conformidade atingida



Fonte: do autor

## 2.2.4 Exemplo de aplicação do *Node-Red* na área da domótica

Com o advento dos equipamentos IOT, a automação residencial conseguiu atingir um público maior, possuindo em seu catálogo diversos dispositivos e soluções para propiciar maior conforto e segurança aos moradores. O exemplo a seguir mostra uma aplicação do *Node-Red* versão 0.20.7 que permite realizar o controle e o monitoramento dos equipamentos presentes em ambientes residenciais.

### 2.2.4.1 Projeto

O objetivo desse projeto é propiciar ao usuário um ambiente no qual seja possível o controle dos equipamentos da residência, permitindo não apenas seu acionamento manual como

também sua programação com o intuito de simular a presença de moradores, fornecendo de mesmo modo um ambiente de monitoramento, no qual se possibilita a visualização de câmeras e o envio de *e-mail* automático em caso de alarme.

Nesse projeto todos os aparelhos podem ser ligados fisicamente ou através da tela de supervisão, tendo seus *status* associados ao *status* real do aparelho. Outra função desse programa é o chamado “Modo viagem”, no qual o usuário pode selecionar o horário que cada aparelho deve ser ligado e desligado, simulando assim a presença de pessoas na residência quando elas não estiverem presentes.

Este sistema possui recurso de envio automático de *e-mail*, que em caso de alarme acionado, este consegue enviar para os endereços cadastrados uma notificação informando o alarme e o momento em que este foi acionado. Ainda é possível ter acesso *on-line* as câmeras instaladas na propriedade, fornecendo ao morador uma forma de monitoramento a distância.

Esse programa dispõe de dois ambientes, um de operação e controle e outro de desenvolvimento. Sendo o primeiro uma interface para que o usuário possa interagir com os equipamentos do sistema, ao passo que o segundo corresponde a um ambiente de programação em fluxo, sendo utilizado para criação ou alteração da estrutura do programa. A seguir será descrito cada ambiente e seus principais elementos.

#### **2.2.4.2 Ambiente de operação**

Este ambiente comporta a tela de supervisão e controle. É nessa tela que o usuário interage com o sistema, controlando equipamentos e configurando rotinas.

##### **Tela de supervisão e controle**

Essa tela (figura 2.34) é dividida em quatro painéis, sendo eles: controle da casa, modo viagem, alarme e monitoramento. A descrição de cada um desses é listada a seguir.

- Controle da casa: nesse painel o usuário pode enviar o comando de ligar ou desligar o aparelho diretamente, esse controle só é habilitado quando a comunicação MQTT estiver ativa, fornecendo ao usuário informações reais do estado do equipamento;
- Modo viagem: possui botão para ativar e desativar o modo viagem, fornecendo também campos para entrada de dados de horário para ligar e desligar os aparelhos configurados;



Figura 2.34 – Tela de supervisão e controle



Fonte: do autor

- Alarme: nesse painel é informado o *status* atual do alarme e se o modo alarme está ativado, nessa parte também é possível configurar o tempo do alarme sonoro e para quais *e-mails* será enviado a notificação. Por fim, este possui dois botões para ativar e desativar o modo alarme;
- Monitoramento: exibe imagens das câmeras previamente cadastradas no sistema, podendo o usuário realizar a seleção dessas por meio da caixa de listagem.

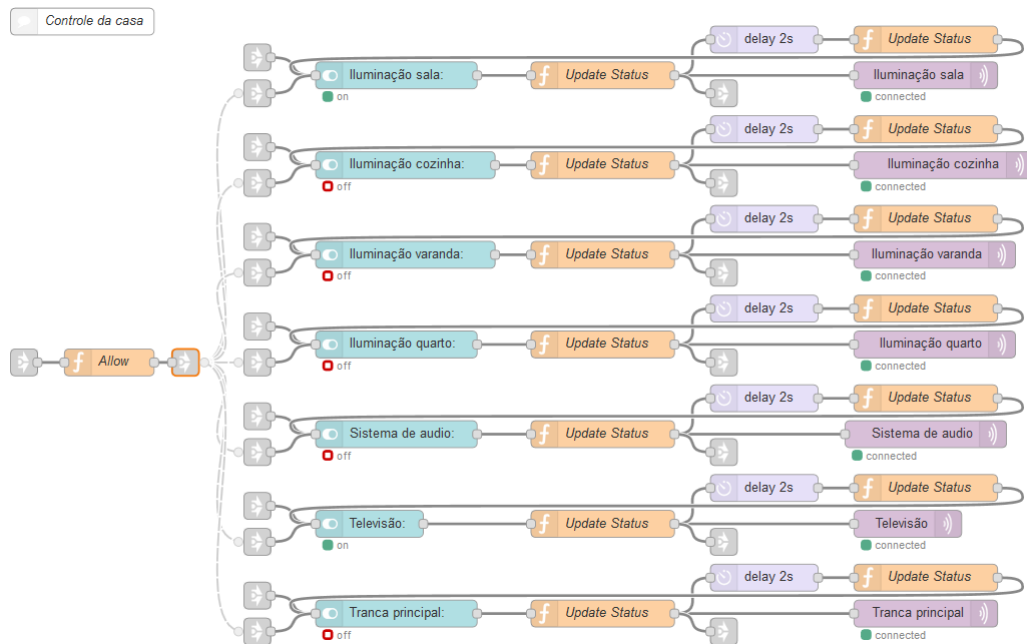
### 2.2.4.3 Ambiente de desenvolvimento

A seguir serão descritos os fluxos que foram utilizados para a criação das lógicas de acionamento, bem como da interface gráfica de supervisão.

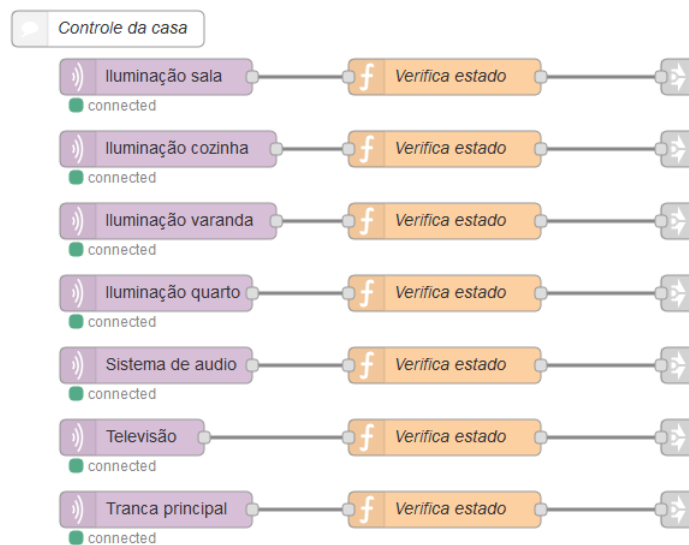
#### Controle dos equipamentos

A figura 2.35 ilustra o fluxo responsável pelo painel de controle dos equipamentos, nele observa-se alguns blocos de funções e temporizações, eles foram utilizados como uma forma de checar se o comando enviado resultou na ação do aparelho, mostrando ao usuário o estado real do dispositivo. A leitura do *status* do aparelho é exibida na figura 2.36. O bloco de função “Allow” tem como atribuição habilitar ou desabilitar os controles em decorrência do *status* do servidor MQTT.

Figura 2.35 – Fluxo responsável pelo controle dos equipamentos



Fonte: do autor

Figura 2.36 – Fluxo responsável pela leitura dos *status* dos equipamentos

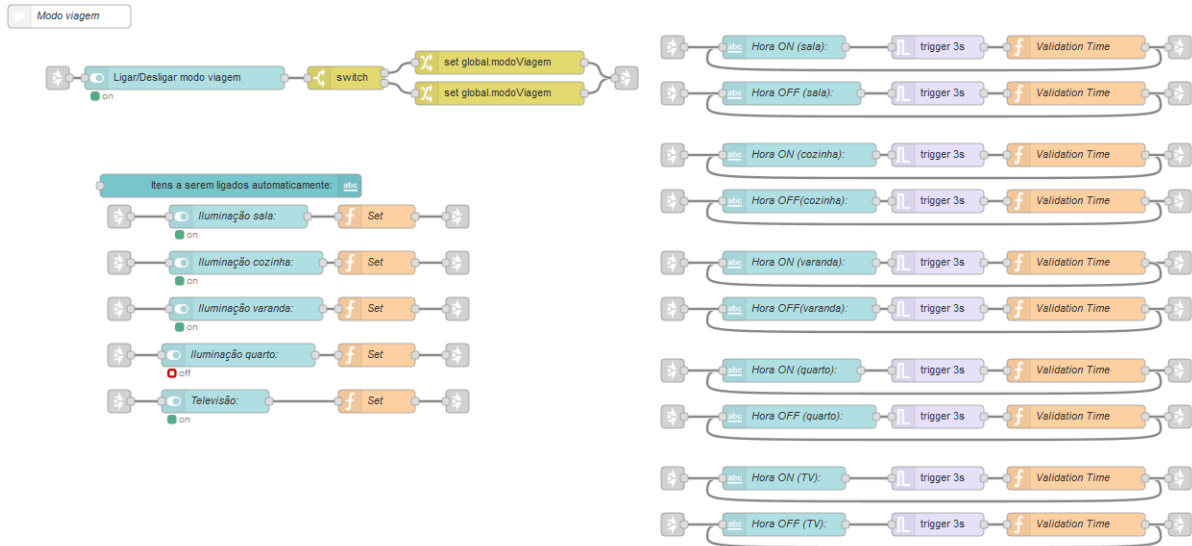
Fonte: do autor

## Modo viagem

A figura 2.37 exibe o fluxo responsável pelo painel “Modo viagem”, nele observa-se a presença de blocos do tipo funções, usados para a gravação dos dados em variáveis globais, bem como, blocos usados para a validação dos dados no formato predeterminado (HH:MM).

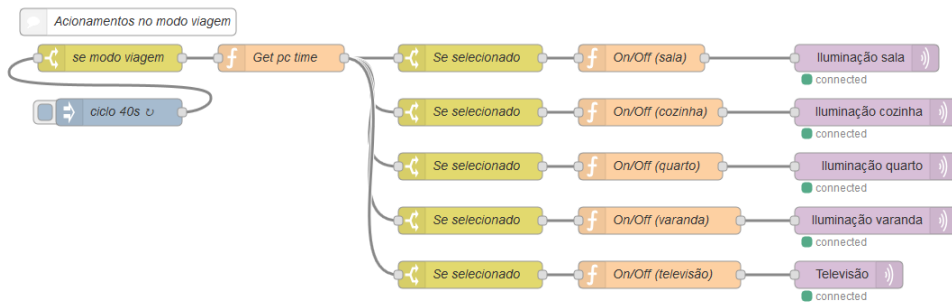
A lógica de acionamento de cada aparelho é ilustrada na figura 2.38, na qual nota-se que o horário do computador é comparado com os horários configurados pelo usuário, que se caso positivo, um comando para ligar ou desligar é enviado aos aparelhos seleccionados.

Figura 2.37 – Fluxo responsável pelo painel modo viagem



Fonte: do autor

Figura 2.38 – Fluxo responsável pela lógica de acionamento do modo viagem

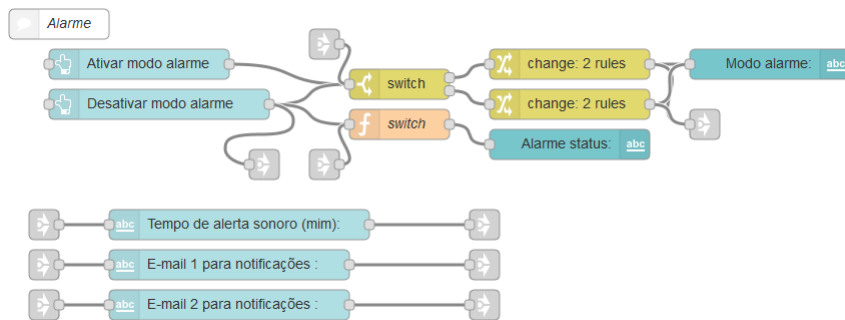


Fonte: do autor

## Alarme

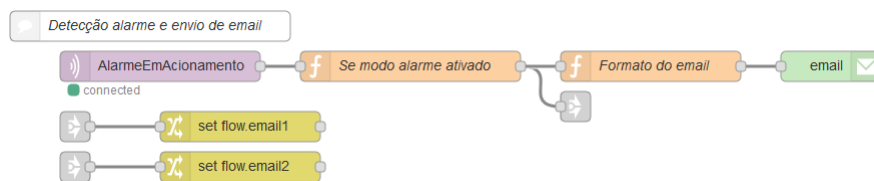
O fluxo exibido na figura 2.39 ilustra a sequência de blocos responsáveis por estruturar e adquirir as informações presentes no painel de alarme. Para realizar o envio do *e-mail* em caso de acionamento do alarme, foi elaborado o fluxo ilustrado na figura 2.40, no qual foi utilizado: uma função para checar o *status* do alarme, um bloco de formatação da mensagem e por fim um bloco de *e-mail*, sendo esse configurado com a conta de *e-mail* escolhida para ser o remetente da mensagem.

Figura 2.39 – Fluxo responsável pelo painel alarme



Fonte: do autor

Figura 2.40 – Fluxo responsável pelo envio de e-mail

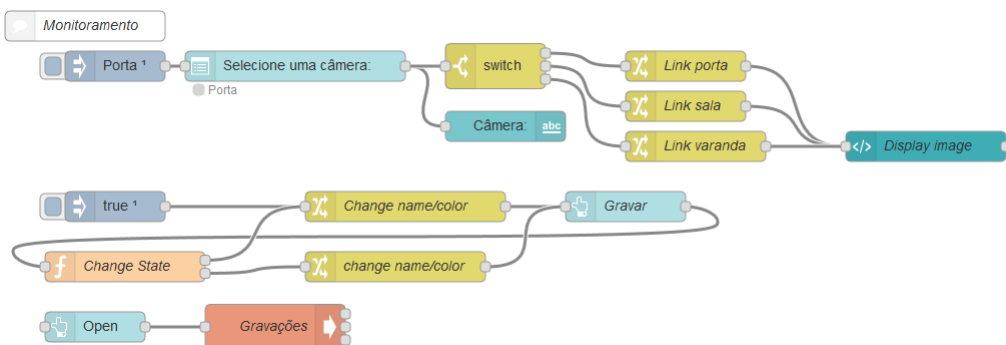


Fonte: do autor

## Monitoramento

O fluxo exibido na figura 2.41 ilustra o arranjo dos blocos responsáveis pela estrutura do painel de monitoramento. Nesse fluxo está presente uma função que altera o estilo e a mensagem do botão gravar, fornecendo informações mais claras ao usuário, melhorando assim a usabilidade do sistema. Nota-se também a presença de um bloco de execução de comandos (bloco “Gravações”) no qual é possível executar comandos do sistema operacional, nesse exemplo ele foi utilizado apenas para abrir o caminho onde os arquivos de gravação estão armazenados.

Figura 2.41 – Fluxo responsável pelo painel de monitoramento

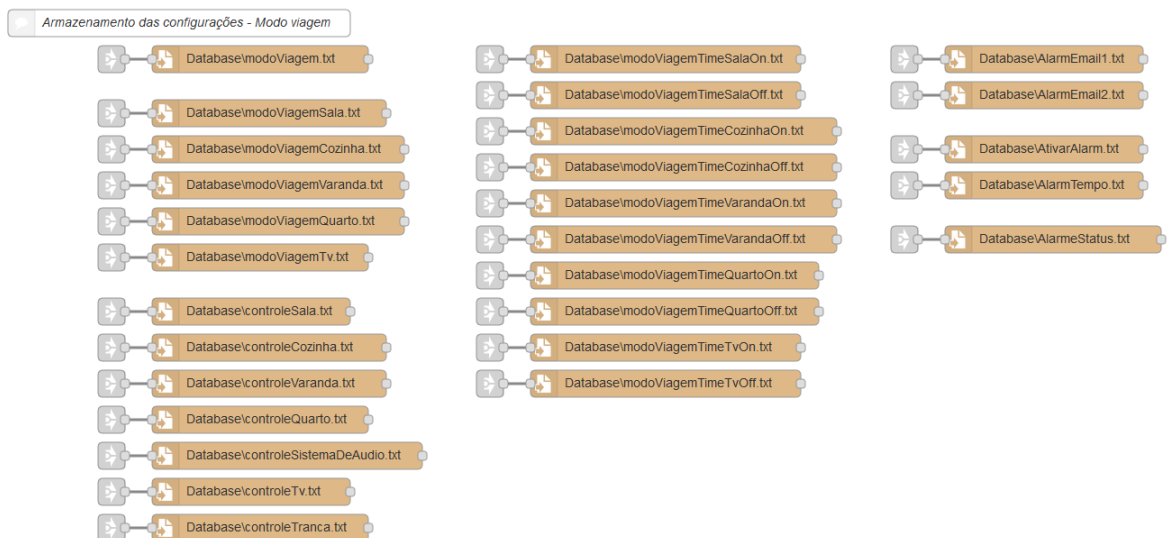


Fonte: do autor

## Inicializações

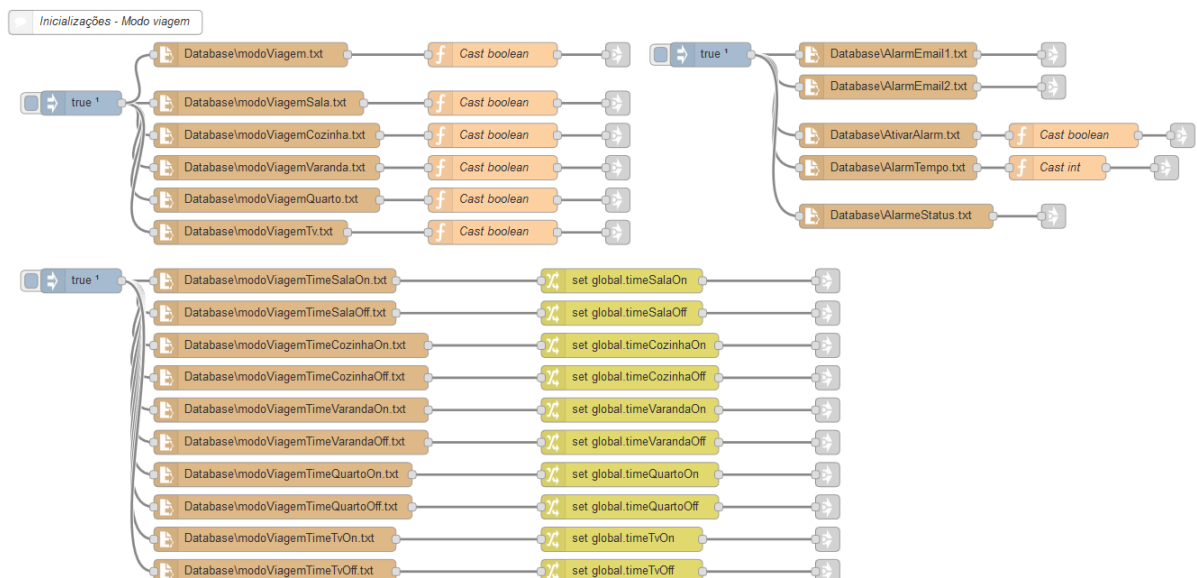
Com o intuito de armazenar todas as configurações previamente realizadas, foram criadas uma série de arquivos de texto, dessa forma, toda alteração é armazenada e carregada automaticamente no início da execução do sistema. As figuras 2.42 e 2.43 exibem os fluxos correspondentes a essas ações.

Figura 2.42 – Fluxo responsável pela gravação das informações do modo viagem



Fonte: do autor

Figura 2.43 – Fluxo responsável pelo carregamento das informações do modo viagem



Fonte: do autor

#### 2.2.4.4 Resultados e simulação

Nesse exemplo foram realizadas as seguintes simulações:

- Programação do modo viagem: foram feitas programações em diferentes horários, verificando assim um comportamento adequado, no qual os equipamentos ligaram e desligaram no tempo estabelecido;
- Controle da casa: foram simulados acionamentos de equipamentos e ausência do serviço MQTT, verificando assim que a lógica de tratamento configurada foi efetiva;
- *E-mail* automático: por meio da simulação de um sinal de alarme, foi confirmado o recebimento do *e-mail* de notificação de alarme no endereço de destino cadastrado;
- Monitoramento: foi utilizada a câmera de um celular juntamente com um aplicativo que consegue criar um endereço IP para acesso remoto, possibilitando assim a exibição do vídeo capturado pelo celular na tela de supervisão e controle.

O comportamento do sistema perante as simulações foi considerado satisfatório, sendo que essas foram fundamentais no desenvolvimento e nas correções de falhas durante a fase de concepção. Assim, o sistema se mostrou eficiente em integrar diferentes ferramentas em uma única tela de operação.

#### 2.2.5 Implementação em *hardware*

De modo a validar as lógicas e o correto funcionamento da aplicação, foi realizada a implementação em *hardware* do exemplo referente ao uso do *Node-Red* no setor agrícola. Possibilitando assim, identificar possíveis erros ou limitações que podem ocorrer quando este é utilizado em ambientes reais. Para tal foram empregados os seguintes elementos:

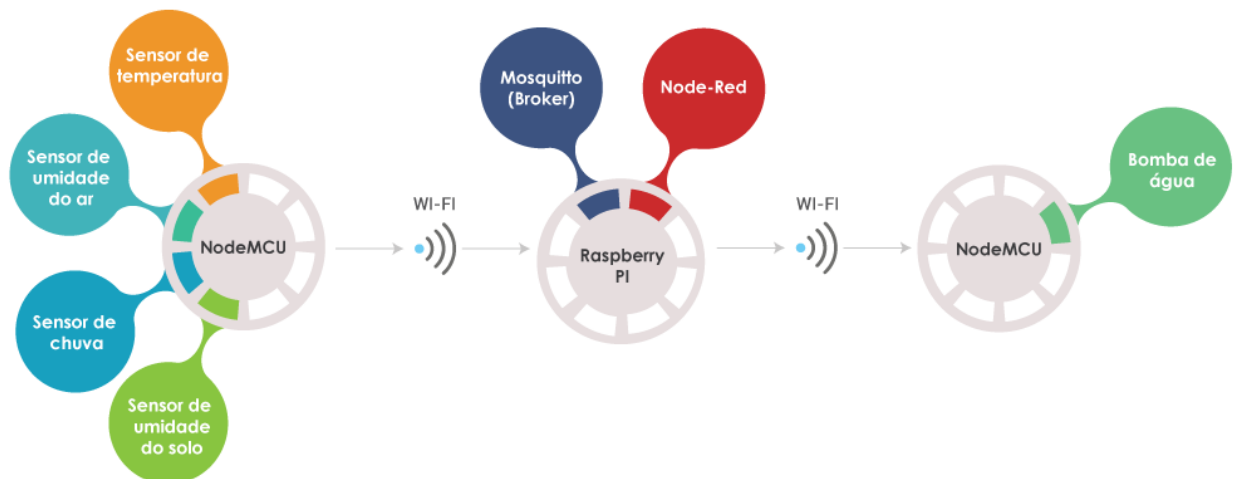
- Raspberry Pi zero: dispositivo no qual a aplicação *Node-Red* e o *broker (Mosquitto)* são executadas;
- NodeMCU: módulo portador do chip ESP8266, utilizado para receber o sinal dos sensores e então enviá-los via protocolo MQTT à aplicação;
- DHT11: sensor de temperatura e umidade, responsável por adquirir o valor de temperatura e umidade do ar, transformando-os em grandezas digitais (Precisão umidade:  $\pm 5\%UR$ ; Precisão temperatura:  $\pm 2^{\circ}C$ );

- Módulo sensor de umidade do solo: utilizado para obter dados referentes a presença de água no solo a ser irrigado (modelo: YL-69);
- Sensor de chuva: empregado para identificar a presença de precipitação no local (modelo: K-0135);
- Bomba de água: elemento atuante usado para irrigar o sistema (modelo: JT100).

A figura 2.44 ilustra os elementos principais pertencentes a essa montagem. Nela observa-se 3 nós, sendo esses responsáveis por realizar a aquisição, processamento dos dados e acionamento do atuador. A seguir são descritas as atribuições de cada nó.

- Aquisição: composto por um NodeMCU que recebe as informações dos sensores instalados no ambiente, enviando esses dados via protocolo MQTT para o nó de processamento;
- Processamento: possui como elemento central um Raspberry Pi zero, no qual executa a aplicação *Node-Red* e um servidor MQTT (*Mosquitto*), dessa forma, a informação obtida por meio dos sensores é então analisada e processada de acordo com as rotinas criadas na aplicação, fornecendo assim, um comando via MQTT para o nó seguinte;
- Acionamento: composto por um NodeMCU que abriga o elemento atuante do sistema (bomba de água), sendo esse operado de acordo com os comandos provenientes do nó de processamento.

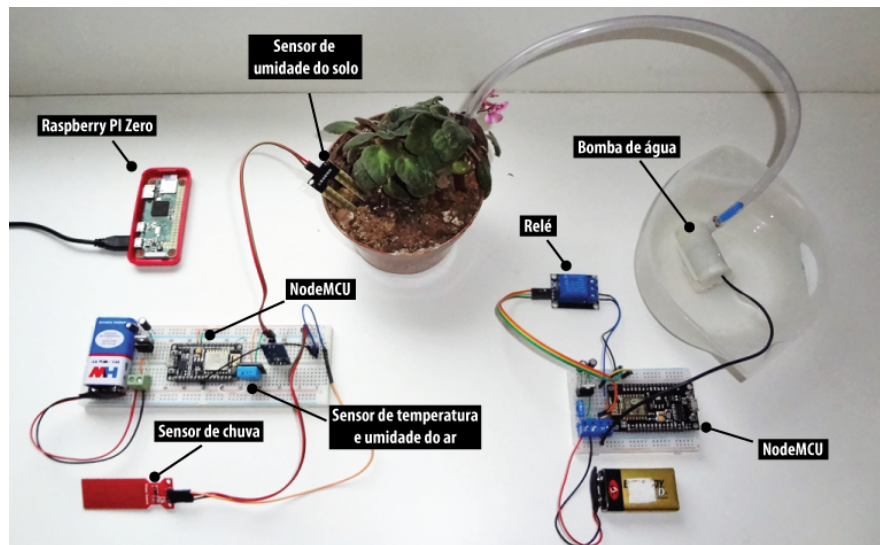
Figura 2.44 – Representação dos elementos pertencentes a implementação em *hardware*



Fonte: do autor

Primeiramente foi realizada a programação dos dispositivos que comportam o chip ESP8266, utilizando para tal, códigos que permitiram sua conexão à rede sem fio local, de igual forma, foram criadas funções para aquisição dos dados dos sensores, sendo esses transmitidos por meio da implementação de bibliotecas para comunicação via protocolo MQTT. A figura 2.45 ilustra os componentes reais utilizados para os testes em questão.

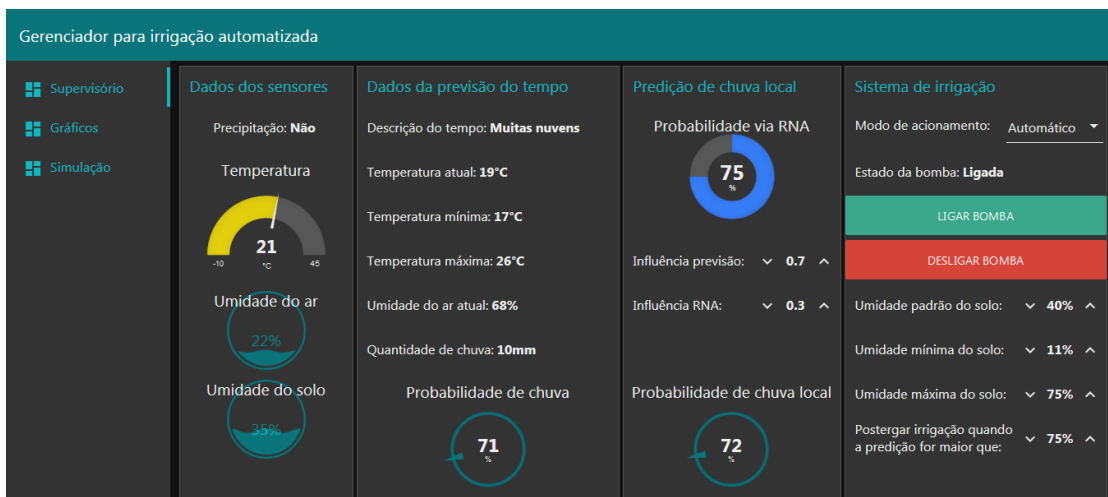
Figura 2.45 – Elementos utilizados na montagem



Fonte: do autor

A figura 2.46 ilustra o resultado da implementação em *hardware* por meio *Dashboard* do sistema de gerenciamento de irrigação, apresentando dados provenientes dos sensores físicos presentes na montagem.

Figura 2.46 – *Dashboard* com dados reais dos sensores



Fonte: do autor



### 2.3 Resultados e discussão

Nesse trabalho foram desenvolvidos três exemplos de emprego da ferramenta *Node-Red* em processos de automatização e controle, abrangendo aplicações que permeiam o setor agrícola, industrial e residencial. Esses projetos foram criados de forma a integrar algum conceito ou ferramenta presente no cenário da quarta revolução industrial dentro de processos tradicionais de automação e controle.

No exemplo referente a aplicação no setor agrícola, foi observado principalmente a facilidade de integrar informações providas da *internet* (previsão do tempo) para auxiliar na tomada de decisão em uma aplicação local. Outra característica interessante presente nesse projeto é observada pela clareza das informações mostradas ao usuário, como dados dos sensores e probabilidade de chuva. Embora apenas exposto, a possibilidade de usar redes neurais artificiais incorporada ao processo, eleva a capacidade da ferramenta para altos níveis de aplicações que abrangem variadas áreas do conhecimento, servindo nesse exemplo como um ajuste fino da previsão de chuva.

A aplicação no setor industrial foi realizada pensando em um processo tradicional dessa área, porém, com alguns conceitos da Indústria 4.0 incorporada, como o caso de integração dos dados dos equipamentos com o fabricante ou empresa responsável pela manutenção, realizando assim de forma automática a solicitação do serviço no momento adequado. Nesse exemplo também foi observado certa facilidade em se trabalhar com operações matemáticas simples, utilizando para tal, funções básicas escritas na linguagem *JavaScript*, permitindo assim, exibir de forma clara e objetiva os dados da produção.

No projeto que abrange a área da domótica, foi obtido uma aplicação capaz de interagir com os dispositivos da residência, entregando maior comodidade e segurança ao usuário. Uma vez que o controle dos equipamentos foi estabelecido via código, a programação desses se mostrou tarefa simples, sendo essa funcionalidade muito útil para simulações de presença. O recurso de envio de mensagem automática em caso de alarme, possui desenvolvimento simplificado e consegue entregar de maneira eficiente a mensagem, podendo essa ser personalizada de acordo com a situação. A incorporação de imagens de vídeo providas de câmeras IP, se estabeleceu de forma simplificada, exigindo apenas duas linhas de código HTML dentro de um bloco nativo ao *Node-Red*.

Com base no desenvolvimento dessas aplicações, foram observados alguns pontos favoráveis e outros pontos de dificuldades, sendo eles:

#### Pontos favoráveis:

- Não necessita profundos conhecimentos de programação para criar aplicações;
- Rápida compreensão da lógica e caminho da informação dentro do programa;
- Capacidade de personalizar a ferramenta e expandir suas funcionalidades;
- Facilidade em criar interface gráfica para supervisão e controle;
- Documentação dos blocos incorporada a ferramenta;
- Fácil integração entre diferentes tecnologias em um único ambiente;
- Capacidade de adicionar programação convencional dentro de blocos;
- Possuir seu desenvolvimento e execução multiplataforma.

#### Dificuldades encontradas:

- Ausência de blocos nativos dedicado a retenção dos estados das variáveis;
- Impossibilidade de personalização de máscaras para validação de entrada de dados;
- Reduzida biblioteca gráfica (chaves, gráficos, botões...);
- Pouca opção de variação no *layout* dos painéis do *dashboard*.

Em resumo, a ferramenta possui recursos nativos que permitem atuar em diversas aplicações, conseguindo entregar bons resultados em curto espaço de tempo, contando com a grande vantagem de ser gratuita e de código aberto, tendo assim seu desenvolvimento contínuo. A maioria dos pontos de dificuldades observados estão atrelados ao fato de esta ser uma ferramenta consideravelmente nova, não possuindo assim, blocos ou paletas já desenvolvidos que atendam todas as necessidades, mas que podem ser criados pela comunidade desenvolvedora ou pelo próprio usuário.

### 3 CONCLUSÃO

Nesse trabalho foram relatadas as principais tecnologias associadas a quarta revolução industrial, mostrando como elas permeiam a sociedade atual e como uma plataforma de programação em fluxo orientada a IOT pode atuar em diferentes processos de automatização que englobem algumas dessas tecnologias.

Foram criados e simulados três exemplos de aplicações, abrangendo os setores agrícola, industrial e residencial, sendo assim observados bons resultados, nos quais a integração entre diferentes tecnologias e conceitos presentes na era atual foram facilmente implementados. Nesses exemplos verificou-se a baixa complexidade requerida para criar programas através do uso da ferramenta *Node-Red*, ficando claro sua importância como ferramenta integradora e facilitadora, fornecendo soluções a baixo custo.

Dessa forma, conclui-se que a ferramenta *Node-Red* consegue atuar de forma satisfatória em alguns processos de automatização e controle, tendo como importante característica sua capacidade de agregar diferentes tecnologias em um único ambiente, por meio de uma interface simples e intuitiva. Por fim, a possibilidade de criar seus próprios blocos e incorporá-los a essa ferramenta, faz do *Node-Red* uma plataforma dinâmica e em constante evolução, sendo assim uma ferramenta alinhada ao cenário da quarta revolução industrial.

## REFERÊNCIAS

- ACROBOTIC. **Módulo ESP8266-12**. 2016. Disponível em: <[www.learn.acrobotic.com/tutorials/post/esp8266-getting-started](http://www.learn.acrobotic.com/tutorials/post/esp8266-getting-started)>. Acesso em: 19 de setembro de 2019.
- DAVIES, E. **Computer Vision: Principles, Algorithms, Applications, Learning**. [S.l.]: Elsevier Science, 2017. ISBN 9780128095751.
- DEANE, P. **The First Industrial Revolution**. [S.l.]: Cambridge University Press, 1979. ISBN 9780521296090.
- FRANCESCHETTI, D. **Principles of Robotics & Artificial Intelligence**. [S.l.]: Salem Press, Incorporated, 2018. (Principles of). ISBN 9781682179420.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [S.l.]: MIT Press, 2016. (Adaptive Computation and Machine Learning series). ISBN 9780262035613.
- HANES, D. et al. **IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things**. [S.l.]: Cisco Press, 2017. (Cisco Press fundamentals series). ISBN 9781587144561.
- ISMAIL, S. et al. **Exponential Organizations: Why new organizations are ten times better, faster, and cheaper than yours (and what to do about it)**. [S.l.]: Diversion Books, 2014. ISBN 9781626813588.
- KITCHIN, R. **The Data Revolution: Big Data, Open Data, Data Infrastructures and Their Consequences**. [S.l.]: SAGE Publications, 2014. ISBN 9781473908253.
- KURZWEIL, R. **The Age of Intelligent Machines**. Cambridge, MA, USA: MIT Press, 1990. ISBN 0-262-11121-7.
- LAMB, F. **Industrial Automation: Hands On**. [S.l.]: McGraw-Hill Education, 2013. ISBN 9780071816472.
- MARR, B. **Big Data in Practice: How 45 Successful Companies Used Big Data Analytics to Deliver Extraordinary Results**. [S.l.]: Wiley, 2016. ISBN 9781119231387.
- MCCULLOCH, W. P. W. S. A logical calculus of the ideas immanent in nervous activity. **Bulletin of Mathematical Biology**, v. 5, p. 115–133, 12 1943.
- MQTT.ORG. **MQTT protocol**. 2019. Disponível em: <<http://mqtt.org/faq>>. Acesso em: 20 de setembro de 2019.
- NODE-RED. **Node-red**. 2019. Disponível em: <<https://nodered.org/about/>>. Acesso em: 17 de setembro de 2019.
- OPENJS. **OpenJs Foundation**. 2019. Disponível em: <<https://openjsf.org/>>. Acesso em: 17 de setembro de 2019.
- OUTMAN, J.; OUTMAN, E. **Industrial Revolution: Almanac**. [S.l.]: UXL, 2003. (Gale virtual reference library). ISBN 9780787665135.
- RASPBERRY PI FOUNDATION. **Raspberry Pi**. 2019. Disponível em: <<https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/>>. Acesso em: 18 de setembro de 2019.

RIELLO, G. **Cotton: The Fabric that Made the Modern World**. Cambridge University Press, 2013. ISBN 9781107000223. Disponível em: <<https://books.google.com.br/books?id=6BON6WWKI4C>>.

RIFKIN, J. **The Third Industrial Revolution: How Lateral Power Is Transforming Energy, the Economy, and the World**. St. Martin's Publishing Group, 2011. ISBN 9780230340589. Disponível em: <<https://books.google.com.br/books?id=vbVELATjyEUC>>.

SCHWAB, K. **The Fourth Industrial Revolution**. [S.l.]: Penguin Books Limited, 2017. ISBN 9780241980538.

SEJNOWSKI, T. **The Deep Learning Revolution**. [S.l.]: MIT Press, 2018. (The MIT Press). ISBN 9780262038034.

SMIL, V. **Creating the twentieth century: technical innovations of 1867-1914 and their lasting impact**. [S.l.]: Oxford University Press, 2005. (Technical Revolutions and Their Lasting Impact). ISBN 9780195168747.

SONKA, M.; HLAVAC, V.; BOYLE, R. **Image Processing, Analysis, and Machine Vision**. [S.l.]: Cengage Learning, 2014. ISBN 9781133593607.

STAFF, D. K. P. et al. **Robot**. [S.l.]: Dorling Kindersley, 2018. ISBN 9780241346754.

STEARNS, P. **The Industrial Revolution in World History**. [S.l.]: Avalon Publishing, 2012. ISBN 9780813347301.

THURSTON, R. **A History of the Growth of the Steam-engine**. [S.l.]: D. Appleton, 1878. (A History of the Growth of the Steam-engine, pts. 1-2).

TRIPATHY, B.; ANURADHA, J. **Internet of Things (IoT): Technologies, Applications, Challenges and Solutions**. [S.l.]: CRC Press, 2017. ISBN 9781351980296.