



KLS

Internet das Coisas

Internet das Coisas

André Márcio de Lima Curvello
Túlio Philipe Ferreira e Vieira

© 2019 por Editora e Distribuidora Educacional S.A.

Todos os direitos reservados. Nenhuma parte desta publicação poderá ser reproduzida ou transmitida de qualquer modo ou por qualquer outro meio, eletrônico ou mecânico, incluindo fotocópia, gravação ou qualquer outro tipo de sistema de armazenamento e transmissão de informação, sem prévia autorização, por escrito, da Editora e Distribuidora Educacional S.A.

Presidência

Rodrigo Galindo

Vice-Presidência de Produto, Gestão e Expansão

Julia Gonçalves

Vice-Presidência Acadêmica

Marcos Lemos

Diretoria de Produção e Responsabilidade Social

Camilla Veiga

Gerência Editorial

Fernanda Miglioranza

Editoração Gráfica e Eletrônica

Renata Galdino

Luana Mercurio

Supervisão da Disciplina

Marcilyanne Moreira Gois

Revisão Técnica

Marcio Aparecido Artero

Marcilyanne Moreira Gois

Dados Internacionais de Catalogação na Publicação (CIP)

Vieira, Túlio Philipe Ferreira

V658i Internet das coisas / Túlio Philipe Ferreira e Vieira, André

Márcio de Lima Curvello. – Londrina:

Editora e Distribuidora Educacional S.A., 2019.

256 p.

ISBN 978-85-522-1687-2

1. IoT. 2. Sensores. 3. Atuadores. 4. Microcontroladores.

I. Curvello, André Márcio de Lima. II. Título.

CDD 005.133

Jorge Eduardo de Almeida CRB-8/8753

2019

Editora e Distribuidora Educacional S.A.

Avenida Paris, 675 – Parque Residencial João Piza

CEP: 86041-100 — Londrina — PR

e-mail: editora.educacional@kroton.com.br

Homepage: <http://www.kroton.com.br/>

Sumário

Unidade 1

Redes de computadores e a Internet das Coisas.....	7
Seção 1	
Introdução à IoT.....	9
Seção 2	
Conceitos básicos de redes aplicados à IoT	28
Seção 3	
Protocolos de comunicação aplicados à IoT.....	44

Unidade 2

Sensores, Microcontroladores e Programação em Internet das Coisas	66
Seção 1	
O Arduino e a IoT	68
Seção 2	
Redes de sensores	90
Seção 3	
Programação em Python como ferramenta para IoT.....	108

Unidade 3

Os dados na nuvem e a IoT.....	130
Seção 1	
O Raspberry Pi e a IoT.....	132
Seção 2	
Computação em nuvem e a IoT.....	150
Seção 3	
Manipulação e análise de dados na IoT.....	167

Unidade 4

Novos desenvolvimentos em IoT	189
Seção 1	
Computação em neblina e computação de borda (nuvens locais).....	191
Seção 2	
Veículos conectados	210
Seção 3	
IoT na prática	226

Palavras do autor

O lá! Seja muito bem-vindo a esta nossa jornada. Durante as próximas unidades deste livro, trilharemos uma longa jornada passando por desafios, descobertas e aprendizados relacionados à Internet das Coisas, ou IoT (do inglês, *Internet of Things*), nosso foco de estudo em busca da otimização e melhoria dos processos, produtos e serviços presentes em nosso cotidiano.

Algumas décadas após o aparecimento do termo computação ubíqua, proposto por Mark Weiser, em 1988, estamos vivenciando a computação pervasiva em todos os ramos da nossa sociedade. Tecnologias que antes não faziam parte de nossas vidas, hoje são indispensáveis em nossos dias. Os smartphones, por exemplo, tornaram-se itens imprescindíveis para a vida moderna. A cada dia a quantidade de dispositivos conectados à internet cresce a um ritmo nunca antes presenciado pela humanidade. Atualmente, temos sensores conectados e interagindo entre si, a fim de realizarem as mais variadas funções. Somando-se esse aumento considerável de dispositivos conectados com o surgimento e amadurecimento de tecnologias mais baratas e de fácil acesso, temos um cenário ideal para o desenvolvimento da Internet das Coisas.

No decorrer desta disciplina, desenvolveremos conceitos básicos e avançados, discutiremos exemplos, conheceremos as principais ferramentas e os principais sistemas embarcados utilizados na IoT, vivenciaremos situações práticas, resolveremos problemas reais e, assim, construiremos todas as bases necessárias para o entendimento e a implementação da IoT. Neste livro, aprenderemos, de maneira gradual e com a apresentação de vários exemplos, todos os pilares para o entendimento deste fascinante mundo da Internet das Coisas. Assim, ao final da nossa caminhada, compreenderemos os conceitos que permeiam o tema IoT, além de propor e implementar soluções de hardware e software que possam ser utilizadas para resolver problemas reais.

Na Unidade 1, iniciaremos com os conceitos mais básicos sobre Internet das Coisas, apresentando alguns exemplos de aplicações reais e mostrando como os sensores e atuadores estão relacionados com a IoT. Além disso, passaremos pelos conceitos fundamentais sobre a arquitetura proposta para a IoT e discutiremos os principais protocolos existentes. Ao final dessa unidade, teremos conhecimento dos conceitos essenciais e dos protocolos mais utilizados para a implementação da IoT. Nesse sentido, o conteúdo desenvolvido na Unidade 1 será fundamental para a compreensão de todas as demais.

Na Unidade 2, em um primeiro momento, serão apresentados os conceitos de redes de sensores de forma aprofundada. Essa análise mais detalhada é necessária, pois as redes de sensores são de grande importância para a construção da IoT. Em um segundo momento, será estudada a primeira placa de prototipagem microcontrolada empregada na IoT, o Arduino. Por último, teremos contato com os conceitos de utilização da linguagem Python. Em resumo, após estudar a segunda unidade, seremos capazes de traduzir a solução de um problema para uma linguagem de programação empregada pela IoT.

Na Unidade 3, conheceremos uma segunda placa de prototipagem empregada na IoT, o Raspberry Pi. Por meio da utilização deste, serão apresentados os conceitos da computação em nuvem aplicada à IoT. A terceira unidade termina com a implementação de uma aplicação que integra sensores, atuadores e computação em nuvem, a fim de consolidar os conceitos de manipulação e análise de dados. Por fim, teremos condição de desenvolver projetos de IoT integrados com plataformas em nuvem.

Na Unidade 4, serão apresentados os conceitos de computação em neblina e computação de borda. Além disso, conheceremos exemplos de várias aplicações de IoT nas mais diversas áreas de conhecimento, com destaque para os veículos conectados. Ao final dessa unidade, você conseguirá empregar todos os conceitos aprendidos, com a intenção de resolver problemas reais vivenciados em seu cotidiano.

A partir disso, mergulharemos neste mundo de descobertas, aprendizados, desafios e conquistas que a Internet das Coisas proporciona. A cada etapa vencida, ficaremos cada vez mais apaixonados com as possibilidades geradas pela IoT.

Unidade 1

Redes de computadores e a Internet das Coisas

Convite ao estudo

Nesta primeira unidade, iniciaremos os nossos estudos sobre a Internet das Coisas. Para compreendermos todo o funcionamento da IoT (do inglês *Internet of Things*), temos que conhecer as principais ferramentas e os principais conceitos que integram e governam o desenvolvimento de aplicações nessa área. Para isso, pavimentaremos a nossa estrada e construiremos as bases necessárias para a concepção de sistemas e circuitos que possibilitem a resolução de problemas reais por meio da IoT.

Na Seção 1.1, apresentaremos algumas aplicações que implementam os conceitos da IoT, bem como alguns dos principais sensores e atuadores utilizados nelas. Além disso, abordaremos as questões de segurança e privacidade dos dados aplicadas à IoT. Em seguida, na Seção 1.2, serão estudadas as redes de comunicação, a partir das principais arquiteturas e da realização de um paralelo entre a comunicação cabeada e sem fio. Conheceremos ainda as ameaças à segurança da IoT e quais são as medidas utilizadas para mitigá-las. Por fim, na Seção 1.3, estudaremos a arquitetura da Internet das Coisas e os protocolos que possibilitam o surgimento e desenvolvimento dessa tecnologia.

Ao final desta unidade, seremos capazes de compreender os conceitos fundamentais que serão recorrentes durante toda a nossa caminhada. Nesse sentido, conheceremos o papel dos principais protocolos de rede e roteamento empregados pela IoT. De posse desse conhecimento, teremos as ferramentas necessárias para compreender o funcionamento da Internet das Coisas. Assim, esta unidade tem um papel primordial na construção do conhecimento e, portanto, os conceitos apresentados serão retomados em vários momentos durante esta disciplina.

Uma grande empresa, que realiza e executa projetos de automação predial e residencial (domótica), está interessada em criar uma casa automatizada que seja referência em automação residencial. Essa moradia será utilizada em campanhas de marketing a fim de atrair novos clientes e demonstrar todas as tecnologias disponíveis no mercado. Você, por ser um estudante dedicado, acaba de ser contratado como estagiário por essa empresa e será o responsável técnico pelo levantamento de requisitos desse grande empreendimento. Os diretores dessa companhia contam com você para criar uma experiência

única para os visitantes dessa casa modelo e, assim, contribuir com a prospecção de potenciais clientes e alavancar o crescimento da empresa.

A casa modelo é dotada de todas as áreas de conveniência existentes em uma residência comum. Ela apresenta quartos, banheiros, cozinha, sala, garagem e escritório. Essa moradia deve ter todas as áreas automatizadas a fim de possibilitar aos moradores maior conveniência, independência para pessoas com mobilidade reduzida, economia de energia e segurança. A automação dessa residência consistirá na adoção de sensores, responsáveis por monitorar o cotidiano dos moradores, e atuadores, que interagirão com o ambiente. Por exemplo, sensores devem indicar caso algum morador esteja entrando ou saindo de um cômodo específico e os atuadores devem, automaticamente, acender ou apagar as luzes. Como pode ser visto, existe a necessidade de conhecer os sensores e garantir que eles comuniquem de maneira satisfatória com os atuadores, sem a necessidade da intervenção humana.

Esperamos que você esteja animado com esta sua nova posição de trabalho. Pense nesse estágio em domótica como um laboratório para conhecer e experimentar os mais variados sistemas e protocolos existentes. Esta é a hora para começar a garantir sua vaga no mercado de trabalho da IoT.

Seção 1

Introdução à IoT

Diálogo aberto

Você sabia que, segundo dados divulgados pela Cisco, em 2008, a quantidade de dispositivos conectados à internet era de, aproximadamente, 6.8 bilhões? E que a população de habitantes da terra apresentava este mesmo número? Esse fato tornou-se um marco no desenvolvimento da IoT, pois, pela primeira vez, para cada ser humano, existia pelo menos um dispositivo conectado à internet. Segundo o mesmo relatório divulgado pela Cisco, atualmente, já existem mais de 50 bilhões de dispositivos ligados à internet, realizando as mais variadas atividades, e esse número tende a crescer nos próximos anos (EVANS, 2011). Sendo assim, existe uma variedade de sensores e atuadores que podem ser empregados em diferentes cenários. Dependendo da aplicação a qual se deseja monitorar e controlar, a escolha dos dispositivos tem impacto direto na qualidade e confiabilidade da rede construída. São os atuadores e os sensores os elementos responsáveis, respectivamente, por sentir e interagir com o ambiente.

Como estagiário, a sua primeira atividade será entregar um relatório dizendo quais sensores e atuadores devem estar presentes em cada um dos cômodos da casa modelo. A união destas ferramentas deve possibilitar aos moradores maior conveniência na realização de atividades cotidianas como acender e apagar a luz de maneira automática, identificar quando está começando a chover e fechar as janelas, abrir a porta da garagem considerando que o veículo do morador esteja próximo, medir a temperatura ambiente e ajustar o ar-condicionado, além da detecção facial para melhorar a segurança.

Além disso, esse relatório deve conter a descrição de uma possível situação em que a utilização desses sensores/atuadores facilitaria a vida ou melhoraria a segurança para os habitantes dessa residência.

Essas são algumas das possibilidades, podendo ser pensadas várias outras que facilitem a nossa vida. Nesse contexto, quais são os sensores e atuadores mais adequados para proporcionar maior comodidade, economia de energia e segurança?

Nesta seção, estudaremos os principais conceitos e casos de utilização de sensores/atuadores, além de entender que os conceitos básicos da IoT são fundamentais para desenvolver, de forma eficiente, aplicações que solucionam problemas reais do nosso cotidiano.

Não pode faltar

Nos últimos anos temos vivido uma transformação sem precedentes na história da tecnologia. A cada dia, milhares de dispositivos são criados e incorporados à internet. Com as premissas de “qualquer coisa”, “em qualquer lugar” e “a qualquer momento”, a Internet das Coisas, ou IoT (do inglês, *Internet of Things*), vem modificando o modo como interagimos com o ambiente e nos relacionamos com a tecnologia.

A Internet das Coisas surgiu após os avanços em alguns campos da ciência como sistemas embarcados, microeletrônica, comunicação e sensoriamento (SANTOS et al., 2016). Com o desenvolvimento da inteligência artificial, várias outras áreas passaram a empregar as redes IoT para a solução de problemas de maneira automática.

A IoT visa criar um mundo inteligente em que o virtual, o digital e o real estejam conectados (DE et al., 2012). A conexão desses diferentes ambientes visa tornar mais eficientes, praticamente em todos os ramos da sociedade, os produtos e serviços oferecidos. Atualmente, encontramos aplicações de IoT que objetivam gerar maior economia de energia, conceber meios de transporte mais eficientes, construir cidades inteligentes, desenvolver indústrias mais sustentáveis, melhorar a qualidade de vida das pessoas, produzir alimentos com maior qualidade e em maior quantidade, além de melhorias em muitas outras áreas da nossa vida.

Saiba mais

Assista ao vídeo do NIC.br (Núcleo de Informação e Coordenação do Ponto BR), uma associação sem fins lucrativos, criada pelos membros do Comitê Gestor da Internet no Brasil, para compreender melhor o conceito de IoT e suas aplicações.

A Internet das coisas, explicada pelo NIC.br. Disponível em: <https://www.nic.br/videos/ver/a-internet-das-coisas-explicada-pelo-nic-br/>. Acesso em: 23 ago. 2019.

Mas o que seria a Internet das Coisas? Esta pode ser vista como um conceito de computação que descreve a conexão de objetos físicos do nosso cotidiano à internet. Por meio da identificação única, a partir do *Internet Protocol (IP)* e do diálogo por meio de *Machine-to-Machine (M2M) Communication*, é possível a comunicação entre esses dispositivos de maneira automática. A IoT corresponde à união de várias tecnologias complementares a fim de possibilitar a interconexão entre o mundo físico e o virtual.

Agora que já definimos IoT, falaremos sobre os elementos que constituem a Internet das Coisas. A união e cooperação desses elementos são os responsáveis por gerar todas as funcionalidades existentes e proporcionadas pela IoT. Existem seis principais elementos que a compõem (AL-FUQAHÁ et al., 2015). São eles:

- **Identificação:** na IoT, existem dois tipos de identificação de dispositivos; o primeiro refere-se ao nome do objeto que é interpretado pelos humanos (sensor de temperatura, atuador 1, ar-condicionado, etc), enquanto o segundo corresponde à identificação de endereço do dispositivo na rede. É por meio desse endereço que as mensagens enviadas pelas redes locais ou integradas via internet conseguem encontrar o destinatário apropriado.
- **Sensoriamento:** na IoT, sensoriamento significa coletar os dados do ambiente e enviá-los para um local no qual possam ser armazenados e, posteriormente, tratados.
- **Comunicação:** a comunicação é um dos principais elementos constituintes da IoT. A IoT é composta por objetos heterogêneos, disponibilidade reduzida de recursos e execução de tarefas nos mais variados ambientes (aquático, terrestre, aéreo, etc). Portanto, prover protocolos robustos e que consigam lidar com as peculiaridades da rede é fundamental para o desenvolvimento da IoT.
- **Computação:** as unidades de processamento e as plataformas em nuvem são os elementos responsáveis por executar as tarefas computacionais dos objetos da IoT. Plataformas de prototipagem, como o Arduino e Raspberry Pi, estão entre as principais ferramentas de desenvolvimento da IoT.
- **Serviços:** existe uma infinidade de aplicações para a IoT, as quais podem ser agrupadas de acordo com o tipo de serviço disponibilizado. Existem quatro diferentes grupos de serviços: serviços relacionados à identificação, serviços de agrupamento da informação, serviços colaborativos e serviços ubíquos. Essa divisão em grupos é utilizada para proporcionar ao desenvolvedor conhecimento sobre os serviços disponíveis em uma aplicação.
- **Semântica:** é a habilidade de transformar os dados coletados em conhecimento. Por exemplo, a partir dos dados vitais de um paciente monitorado (pressão arterial, frequência cardíaca e níveis de glicose no sangue), é possível gerar o mapeamento da saúde desse indivíduo em tempo real.

Você já pensou que, atualmente, nós podemos transformar, praticamente, qualquer objeto do nosso cotidiano em um elemento pertencente à IoT? Para tanto, precisamos garantir os requisitos mínimos que possibilitem a esse objeto interagir com o ambiente de maneira autônoma. Dessa forma, quatro partes principais devem ser implementadas (FORTINO; TRUNFIO, 2014):

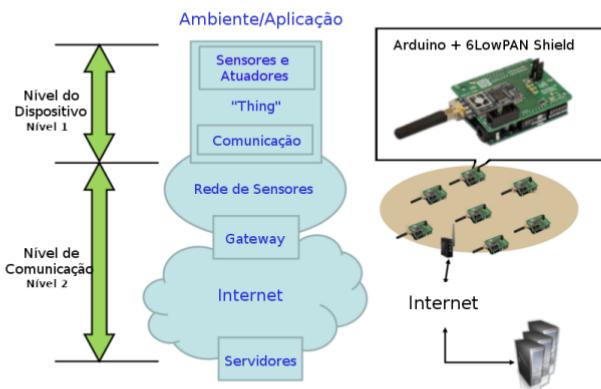
- **Sensores e atuadores:** são responsáveis pela interação com o ambiente físico. Os sensores captam informação do ambiente e os atuadores têm a responsabilidade de modificá-los.

- **Unidade de processamento:** é responsável por realizar o processamento dos dados e coordenar as operações dos demais sistemas.
- **Unidade de comunicação:** é por meio dessa unidade que ocorre a troca de informação entre os elementos constituintes da IoT. Em síntese, a unidade de comunicação é responsável por criar a rede de dispositivos.
- **Fonte de energia:** como todo elemento precisa realizar tarefas, a fonte de energia é vital para todos os dispositivos integrantes da IoT.

Assim, quase todos os dispositivos pertencentes à IoT compartilham características que os tornam distintos dos elementos tradicionais. Quando esses dispositivos não realizam funções críticas como em veículos autônomos, normalmente utilizam baterias como fonte de energia, apresentando, portanto, uma fonte limitada de recursos. Esses elementos, usualmente, têm pequenas dimensões e são construídos para realizarem tarefas com reduzida complexidade. Quando a aplicação não requer alto nível de confiabilidade e disponibilidade, a alimentação desses dispositivos é realizada por baterias. Devido à limitação de energia, as questões de desempenho e comunicação devem priorizar o consumo eficiente de recursos energéticos (baterias). Outra característica dos elementos da IoT é o baixo preço de produção. Dessa forma, ao dotarmos os objetos do nosso cotidiano dessas quatro partes fundamentais, estamos criando os sistemas embarcados que construirão a Internet das Coisas.

A IoT é um sistema complexo que integra diferentes plataformas de hardware e software dos mais variados fabricantes, desenvolvendo diversas funções que devem comunicar entre si por meio da internet (HERSENT *et al.*, 2011). Para auxiliar no entendimento, a Figura 1.1 apresenta uma visão geral sobre o funcionamento dos hardwares da IoT.

Figura 1.1 | Visão geral dos componentes de hardware na IoT



Fonte: adaptada de Hersistent *et al.* (2011).

A partir da análise da Figura 1.1, encontramos a divisão em dois níveis. O primeiro corresponde ao nível do dispositivo, que pode ser visto como a parte do hardware responsável por interagir com o ambiente, na qual é possível encontrar os sensores e atuadores. O segundo refere-se ao nível de comunicação entre os dispositivos. Essa troca de mensagens pode acontecer tanto dentro de uma rede local quanto entre redes geograficamente distantes por meio da internet. Para que essa comunicação ocorra de maneira correta, é necessário que sejam utilizados gateways, isto é, elementos (hardware ou software) que têm a função de transportar a informação entre os dispositivos IoT e a internet e, assim, possibilitar que as mensagens sejam compreendidas em ambos os lados (HERSENT *et al.*, 2011).

Exemplificando

Os gateways apresentam papel fundamental não apenas para a Internet das Coisas, mas também para outras áreas da computação. Eles têm a função de organizar a troca de mensagens entre os dispositivos finais (computadores, smartphones, tablets, dispositivos IoT, etc.) e a internet. Além disso, são responsáveis por “traduzir” essas mensagens trocadas e, assim, possibilitar que a informação seja compreendida por qualquer dispositivo que deseje participar da conversa. Um exemplo de gateway é o roteador de internet, geralmente instalado pela operadora de telefonia, que garante que as mensagens enviadas possam percorrer todo o planeta e sejam corretamente interpretadas por qualquer computador, tablet ou smartphone ao redor do mundo (TANENBAUM, 2003).

Aplicações em IoT

Agora que já conhecemos os principais conceitos necessários ao entendimento da Internet das Coisas, podemos apresentar alguns exemplos de aplicações desse sistema. Os carros autônomos representam uma das mais futurísticas aplicações da IoT. Nesse tipo de aplicação, automóveis comuns são modificados e milhares de sensores como acelerômetros, giroscópios e velocímetros são integrados aos atuadores, aos sistemas de *Global Positioning System (GPS)* e à internet, todos eles utilizados para monitorar as condições da via, a movimentação dos pedestres e dos outros veículos, realizar conversões e interagir com todo o sistema de trânsito, a fim de possibilitar a realização de viagens sem a necessidade de um motorista. Para os carros autônomos, a capacidade de processamento deve ser bastante elevada, pois as respostas a eventos devem ser obtidas em frações de segundos (MAHMOOD, 2019). Outra característica dos veículos autônomos é a presença massiva de algoritmos de inteligência artificial, empregados para auxiliar na tomada de

decisão em situações de maior criticidade. Não só grandes empresas de tecnologia, mas também grandes montadoras já apresentam carros autônomos e investem nesse produto. A Figura 1.2 apresenta um exemplo de projeto de carros autônomos idealizado pela Google.

Figura 1.2 | Exemplo de um veículo autônomo desenvolvido pela Google



Fonte: Jurvetson (2012).

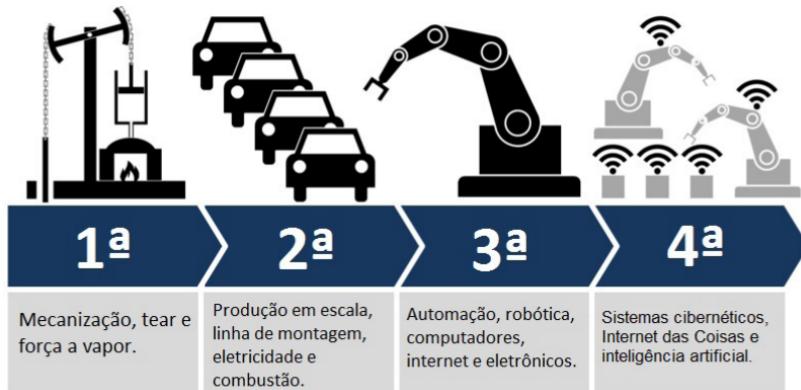
Outra aplicação da Internet das Coisas que certamente mudará a forma como utilizamos o espaço urbano é o conceito de cidades inteligentes, que consistem em espaços urbanos que utilizam a IoT para aperfeiçoar a infraestrutura urbana, tornar os produtos e serviços mais eficientes e, portanto, gerar melhor qualidade de vida para todos os moradores das cidades (DAMERI *et al.*, 2017). No desenvolvimento das cidades inteligentes, questões relativas à mobilidade, ao meio ambiente e à melhoria no bem-estar geral dos cidadãos são os pilares dessa aplicação. Na República de Singapura, por exemplo, vários sensores foram espalhados pelo local. Esses sensores conseguem detectar se existe algum cidadão fumando em local proibido e identificar o indivíduo que estiver despejando lixo pela janela dos edifícios. Como você pode ver, a IoT possibilitou a realização de tarefas que eram exclusivas aos filmes de ficção (SANCHEZ *et al.*, 2014), (GAUR *et al.*, 2015), (KIM *et al.*, 2017).

Caso ainda não esteja convencido sobre o potencial transformador da IoT e que este é um tema que deve estar presente em seus pensamentos, apresentaremos mais dois exemplos de aplicações. O primeiro é

na medicina, uma área em que a IoT contribui bastante para a melhoria da nossa sociedade. Dispositivos IoT são empregados para o monitoramento de pacientes com câncer, no acompanhamento da glicose em pessoas com diabetes e aplicação automática de insulina, além do desenvolvimento de um pâncreas artificial (LEWIS; LEIBRAND, 2016). Outra aplicação que parece ter saído dos filmes de ficção é a utilização de pílulas, que, ao se dissolverem no estômago, geram sinais elétricos interpretados por sensores presentes no corpo do paciente. Com a introdução dos SmartWatch, já é possível monitorar os primeiros sinais da doença de Parkinson e detectar os sintomas que antecedem os ataques de asma e, assim, possibilitar que o paciente consiga gerenciar de maneira mais adequada a doença (COHEN *et al.*, 2016).

O último exemplo de aplicação de IoT é encontrado no desenvolvimento da indústria 4.0, termo utilizado para indicar as transformações geradas pela adoção da IoT na indústria moderna (GILCHRIST, 2016). O emprego cada vez maior da interconectividade, automação, aquisição e análise de dados em tempo real no ambiente industrial proporciona o surgimento da indústria 4.0. O sistema industrial completamente integrado, colaborativo e que responda, em tempo real, às mudanças exigidas é o principal preceito da indústria 4.0. Com a adoção desse novo paradigma, as empresas poderão oferecer serviços personalizados com a mesma facilidade que produzem em uma linha de produção. Assim, os produtos e serviços podem ser gerados com qualidade superior, maior velocidade, menor gasto de energia e com preços reduzidos. A Figura 1.3 mostra um resumo do desenvolvimento industrial até o aparecimento da indústria 4.0 (GILCHRIST, 2016).

Figura 1.3 | Desenvolvimento industrial até o surgimento da indústria 4.0



Fonte: adaptada de Roser (2015).

Como você já deve ter percebido, todas as áreas em nosso cotidiano empregam, com algum grau de utilização, as premissas da Internet das Coisas para melhorar a eficiência dos processos e produtos.

Os elementos fundamentais que possibilitam a construção de todas as aplicações de IoT são os sensores e atuadores. Conhecer os diferentes tipos e como eles funcionam são as premissas para desenvolver aplicações em IoT.

Sensores

Os sensores são os elementos conhecidos como entradas da IoT, visto que são responsáveis por gerar os dados para uma aplicação. Além disso, eles são capazes de medir as variáveis do ambiente (temperatura, pressão, umidade e velocidade), recebendo estímulos físicos deste e respondendo por meio de sinais elétricos, convertidos em grandezas físicas relacionadas à variável medida (HANES *et al.*, 2017). Um sensor de temperatura do tipo LM35, por exemplo, quando submetido a uma variação de temperatura no ambiente, promove uma variação da tensão de saída em seus terminais. Essa variação de tensão pode ser medida e, por meio de equações de transformações, pode-se encontrar um valor de temperatura medido (INSTRUMENTS, 1999).

Para cada elemento sensor existe uma função de transferência associada, isto é, uma equação matemática responsável por representar a relação entre a entrada e a saída. A entrada da função de transferência para um sensor corresponde ao parâmetro físico medido no ambiente. Já a saída para um sensor, mapeada pela função de transferência, refere-se a um sinal elétrico. No caso do exemplo do LM35, a entrada é o valor da temperatura ambiente, enquanto a saída representa a tensão gerada nos terminais do sensor. Para o LM35, a função de transferência é a seguinte:

$$V_{out} = 10 \frac{mV}{^{\circ}C} T$$

Onde: V_{out} = tensão de saída do LM35

T = temperatura em $^{\circ}C$

Analizando essa função de transferência, chegamos a seguinte relação: cada 10mV gerados na saída do LM35 correspondem à variação de $1^{\circ}C$. Assim, se na saída é medido 222 mV, podemos assumir que a temperatura medida pelo LM35 será igual a 22,2 $^{\circ}C$.

Assimile

Ao utilizar os sensores, é comum ouvirmos o termo transdutor. Em algumas situações esses dois termos são empregados, de maneira equivocada, como sinônimos. Transdutores e sensores são elementos diferentes. Os sensores são utilizados para detectar variações no ambiente, enquanto os transdutores convertem um sinal/grandeza de entrada em outro sinal/grandeza de saída. Os transdutores, normalmente, possibilitam que a informação sentida pelo sensor seja convertida em um sinal e que possa ser interpretada pela unidade de processamento.

Agora que já conhecemos as características de um sensor e como ele é utilizado para monitorar o ambiente, apresentaremos vários diferentes tipos de sensores para que você possa escolher os ideais para seus projetos.

Sensores de umidade e temperatura: o sensor mais famoso utilizado para medir a temperatura, por meio de contato, é o LM35. Os sensores conhecidos como DHT11 e DHT22 são elementos que apresentam a capacidade de medir a temperatura e a umidade do ar em um mesmo equipamento (GAY, 2018).

Sensores de proximidade: são classificados segundo o tipo de tecnologia utilizado para medir distâncias em indutivo, capacitivo e ultrassônico. Os sensores induktivos fazem uso do eletromagnetismo para detectar a proximidade de elementos metálicos, sendo utilizados apenas na detecção desses tipos de elementos. Os sensores capacitivos de proximidade podem ser utilizados tanto com objetos metálicos quanto não metálicos, sendo a sua ação gerada pela variação da capacidade entre um objeto e o sensor. Os sensores de proximidade óptica podem ser utilizados para detectar objetos metálicos ou não, além de medir as distâncias e partem do princípio de funcionamento baseado na propagação da luz. Por último, temos os sensores ultrassônicos; devido à diferença de tempo entre uma onda emitida, refletida e recebida pelo receptor, é computada a distância entre o objeto e o sensor.

Acelerômetros: os acelerômetros estão presentes em quase todos os smartphones e tablets existentes. Eles não só indicam a orientação da tela que estamos usando, como também estimam a velocidade. No que se refere ao seu princípio de funcionamento, este é baseado no sistema conhecido como massa e mola, governado pela Lei de Hooke.

Sensores Passive Infrared (PIR) ou sensores de movimento: detectam a presença de seres humanos ou animais no ambiente. Seu princípio de funcionamento é baseado na premissa de que todo corpo que tem temperatura emite

energia térmica na forma de radiação. Assim, o PIR consegue captar essa luz infravermelha e indicar a presença de animais ou seres humanos em ambientes.

Sensor de gás: detecta vários tipos de gases, sendo o MQ-2 o sensor mais utilizado, empregado para detectar gases combustíveis e fumaça (HARIYAWAN *et al.*, 2013).

Sensor de infravermelho: esse sensor tem um LED emissor de infravermelho e um fotodiodo que pode receber radiação infravermelha. É utilizado para receber sinais de controle remoto.

Sensores de nível: monitoram a variação de níveis em diferentes líquidos, podendo ser de vários modelos diferentes (capacitivo, ultrassônico, micro-ondas, etc.).

Sensores de luminosidade: medem a incidência de luz, sendo o fotore sistivo *Light Dependent Resistor (LDR)* o mais utilizado. Esse sensor varia sua resistência devido à intensidade de luz incidente (JUNIOR; JUNIOR, 2013).

Barômetros: medem a pressão atmosférica, sendo o BMP180 o mais conhecido, que também mede a temperatura ambiente. Por meio desses dados, é possível inferir a altitude do local no qual está localizado o sensor.

Magnetômetros: medem a intensidade, a direção e o sentido dos campos magnéticos ao redor, podendo ser utilizados em pesquisas climáticas, identificação em radiocomunicação ou localização (bússola).

Medidores de consumo de energia: para medir o consumo de energia elétrica, a potência elétrica consumida e o tempo em que ela é gasta devem ser conferidos. Para isso, é necessário monitorar a tensão e a corrente durante um período. Para medir essas variáveis, podem ser utilizados amperímetros do tipo alicate e, a partir disso, estimar o consumo de energia de um equipamento (BALBINOT; BRUSAMARELLO, 2011).

Agora apresentaremos outra classe de elemento muito importante: os atuadores.

Refita

Vimos aqui que os dispositivos da IoT apresentam limitada fonte de recursos, reduzido poder de processamento e tamanho diminuto. Todas essas características são decorrentes da fonte de energia escassa que, normalmente, é fornecida por baterias. Você já parou para pensar como os nós de rede IoT, mesmo com recursos limitados, conseguem realizar toda essa transformação que vivenciamos? Você saberia dizer se esses

dispositivos utilizam o mesmo protocolo IPv4 dos computadores? Ou será o IPv6? Ou nenhum dos dois?

Atuadores

Depois de conhecer os principais sensores empregados em projetos de IoT, podemos começar a falar em como modificaremos o ambiente de maneira automática. Para isso, utilizaremos os atuadores, considerados os elementos responsáveis pelas saídas em uma aplicação IoT e por realizar alguma modificação no ambiente.

É através dos atuadores que as ações são executadas no ambiente, ou seja, são eles que executam as interações com o meio. Acender ou apagar uma luz, abrir ou fechar uma janela ou válvula, reproduzir algum som, exibir alguma mensagem, mover objetos e gerar vibrações são apenas alguns dos exemplos de interações possíveis (HANES *et al.*, 2017).

Existem várias classificações para os atuadores, podendo eles serem classificados quanto ao tipo de energia de saída, pelo princípio de funcionamento ou pelo tipo de movimento gerado. Para o nosso estudo, teremos a seguinte classificação: hidráulico, pneumático ou elétrico (CIRINO, 2018).

- Os atuadores hidráulicos utilizam potência hidráulica para realizar uma ação. Esse tipo de atuador pode trabalhar com grandes cargas, mas as respostas geradas são lentas.
- Os atuadores pneumáticos utilizam ar comprimido para gerar uma ação. Quando comparado com os atuadores hidráulicos, apresentam maior velocidade de resposta.
- Os atuadores elétricos fazem uso da eletricidade para realizar uma ação. São os mais versáteis e representam os tipos de atuadores mais empregados em aplicações de IoT, uma vez que apresentam respostas rápidas e trabalham em uma grande faixa de carga.

Depois de compreender o conceito de cada um dos atuadores, iniciaremos o estudo sobre um dos mais importantes atuadores elétricos para realizar a interação com o ambiente: os **motores elétricos**. Eles são utilizados em bombas hidráulicas e ventiladores, além de serem empregados na movimentação de cargas, na construção de robôs e em várias outras aplicações. São considerados motores elétricos os motores de corrente contínua e de passo. Os motores de corrente contínua são compostos por um rotor e um estator, responsável por gerar um campo magnético. Quando as bobinas do rotor são percorridas por uma corrente elétrica e submetidas ao campo gerado pelo estator, devido às leis do magnetismo, surge um torque que gira

o eixo do rotor; essa rotação é utilizada para movimentar as cargas. Exemplos de aplicações desses motores são robôs, esteiras robóticas e movimentadores. Os motores de passo, geralmente, são utilizados quando precisamos posicionar cargas com maior precisão. Isso ocorre pois, devido ao posicionamento através de ângulos de rotação (passos), a variação do deslocamento é mais precisa (PETRUZELLA, 2013). Além desses, existem outros motores elétricos, por exemplo, o de indução, o sincronizado e o universal.

Além dos motores, temos os **atuadores lineares**, as **válvulas solenoides** e as **chaves solenoides**. Os **atuadores lineares** são como hastes móveis que exercem força sobre algum objeto por meio de uma trajetória linear. Esses atuadores podem ser mecânicos, pneumáticos ou elétricos. Estes últimos podem ser utilizados, por exemplo, para abrir e fechar uma janela.

As **válvulas solenoides** são muito empregadas para controlar a passagem de algum fluido. Ao acionar uma válvula solenoide, é gerado um campo magnético que puxa o êmbolo contra a ação de uma mola. Quando está desenergizado, esse êmbolo volta à posição normal, podendo, assim, liberar ou restringir a passagem do fluido. As **chaves solenoides** apresentam o mesmo princípio de funcionamento das válvulas, mas são utilizadas para liberar ou travar a passagem de algum indivíduo.

Atuadores que podem exercer a função de chaves elétricas também são muito utilizados. Os **relés** e os **transistores** podem ser empregados para o acionamento de cargas elétricas como interruptores de lâmpadas e abertura e fechamento de portas.

Outros atuadores bastante recorrentes são os **alarmes**, que têm um papel fundamental para a IoT, principalmente, em questões de segurança, pois é por meio deles que o sistema consegue informar os usuários quando algum problema está ocorrendo. Existe uma infinidade de atuadores, devendo a escolha desses elementos ser realizada por meio da análise da aplicação, pois cada atuador apresenta características específicas que o tornam mais apto a realizar determinada tarefa (DOUKAS, 2012).

Para terminar, vale salientarmos que nesta seção tivemos contato com os conceitos iniciais sobre IoT, bem como vários sensores e atuadores empregados em aplicações de automação. Esperamos que você tenha percebido que nossa jornada ainda é muito longa, mas que será bastante recompensadora. Esta seção inicial é, normalmente, mais teórica, pois precisamos sedimentar conceitos que serão as bases para o decorrer da disciplina. Nesse sentido, começaremos, na próxima unidade, a implementar a IoT.

Sem medo de errar

Como estagiário, recém-contratado por uma empresa de domótica, você recebeu a importante missão de realizar um levantamento prévio sobre quais sensores e atuadores devem ser utilizados para construir a casa modelo em automação. Você deve entregar esse levantamento em forma de um relatório. Essa casa contém quartos, cozinha, banheiro, sala, garagem e escritório. Todos esses cômodos devem ser automatizados a fim de proporcionar mais comodidade e segurança aos moradores. Chegou a hora de mostrar aos diretores que essa tarefa foi executada com maestria.

Como estudamos nesta seção, os sensores e atuadores são as “entradas” e “saídas”, respectivamente, de uma aplicação em IoT. Sendo assim, têm papel fundamental no desenvolvimento de aplicações eficientes e a escolha adequada desses elementos garante maior conforto, economia e praticidade para os usuários dessa casa modelo.

A partir disso, é necessária a apresentação de um relatório contendo as especificações de sensores e atuadores que devem estar presentes em cada um dos cômodos da casa modelo em automação. Além disso, você deve apresentar alguma aplicação em domótica para os sensores/atuadores escolhidos. Nesse contexto, as Quadro 1.1 e 1.2 apresentam, respectivamente, algumas opções de atuadores e sensores para cada um dos cômodos da residência.

Quadro 1.1 | Tabela de indicação de sensores a serem utilizados

Ação	Atuadores					
	Banheiro	Quartos	Porta de Entrada	Cozinha	Escritório/sala	Garagem
Acender/apagar as luzes/equipamentos	Chaves relé	Chaves relé	X	Chaves relé	Chaves relé	Chaves relé
Fechar/abrir janelas	Motores de passo ou atuadores lineares	Motores de passo ou atuadores lineares	X	Motores de passo ou atuadores lineares	Motores de passo ou atuadores lineares	Motores de passo ou atuadores lineares
Abertura/fechamento das portas	X	X	Chaves solenoides	X	X	X
Notificação de invasores	X	X	Alarms sonoros	X	X	X
Ajuste da temperatura	Resistores variáveis (potenciômetros digitais)	X	X	Resistores variáveis (potenciômetros digitais)	X	X

Fonte: elaborado pelo autor.

A seguir, são listadas algumas possíveis aplicações para cada um dos cômodos da residência.

Quadro 1.2 | Tabela de indicação de sensores a serem utilizados

Ação	Sensores					
	Banheiro	Quartos	Porta de Entrada	Cozinha	Escritório /sala	Garagem
Identificar a presença de moradores	Sensores PIR	PIR	X	PIR	PIR	PIR
Identificar chuva	Sensor de umidade capacitivo	Sensor de umidade capacitivo	X	Sensor de umidade capacitivo	Sensor de umidade capacitivo	Sensor de umidade capacitivo
Identificar fumaça/elemento inflamável	Sensores MQ-2	Sensores MQ-2	Sensores MQ-2	Sensores MQ-2	Sensores MQ-2	Sensores MQ-2
Medir a intensidade luminosa	Sensores LDR	Sensores LDR	X	Sensores LDR	Sensores LDR	X
Monitorar o consumo de energia	Alicate amperímetro	Alicate amperímetro	X	Alicate amperímetro	Alicate amperímetro	X
Identificação de pessoas	X	X	Câmeras de vigilância e sensores PIR	X	X	Câmeras de vigilância e sensores PIR
Identificação do veículo	X	X	X	X	X	Câmeras de vigilância e RFID
Aproximação do veículo	X	X	X	X	X	Sensores ultrassônicos e indutivos
Medidores da qualidade do ar	X	X	X	X	X	Sensores de CO e CO ₂
Medir a temperatura da água	Termômetro	X	X	Termômetro	X	X

Fonte: elaborado pelo autor.

Quartos

Os sensores e atuadores presentes no quarto podem ser combinados a fim de, por exemplo, identificar quando os moradores estão em seus dormitórios. A partir disso, os sensores LDR podem monitorar a luminosidade e enviar sinais para ajustar as cortinas dos quartos. Caso os moradores deixem os dormitórios e esqueçam algum equipamento eletrônico ligado, é possível que os alicates amperímetros identifiquem este consumo e acionem os relés para desligar os equipamentos.

Porta de entrada

Para uma maior segurança, é possível que exista uma combinação de autenticações para liberar a entrada. Por exemplo, pode ser que seja necessária a validação de acesso por meio do reconhecimento facial e das digitais. Assim, o acesso torna-se mais restrito e seguro.

Cozinha

Detectores de fumaça podem ser combinados com alarmes sonoros e câmeras de vigilância. Assim, quando um possível foco de incêndio é identificado, a câmera pode ser acionada com um alarme sonoro. Portanto, uma mensagem é enviada para os usuários cadastrados e o acesso às câmeras é liberado. Dessa forma, é possível que essas pessoas cadastradas tenham acesso às imagens das câmeras em tempo real, auxiliando as autoridades no resgate.

Garagem

É possível colocar alguns *tags* de RFID para identificar, de maneira única, os veículos da residência. Para isso, é necessário que exista, por exemplo, um *tag* passivo nos automóveis e um *tag* leitor (ativo) na porta da garagem. Quando um sensor ultrassônico identificar que um carro está se aproximando, ele poderá acionar o leitor ativo de RFID para identificar se o veículo é conhecido. Caso seja afirmativo, os motores do portão serão acionados, liberando a entrada de maneira automática.

Banheiro

É possível que os moradores apresentem temperaturas para o banho predefinidas. Assim, quando a temperatura ambiente é alterada, devido às mudanças das estações do ano, por exemplo, os potenciômetros poderão ajustar, de modo automático, a temperatura ideal para um banho mais confortável.

Essa é apenas uma possível solução para a situação-problema apresentada. Como este é somente um exercício para despertar a curiosidade e exemplificar as possibilidades geradas pela IoT, você pode pensar em várias outras que se tornariam mais eficientes e cômodas por meio desse sistema. Quais outras melhorias você pode sugerir para os diretores? Quais sensores e atuadores são necessários?

Avançando na prática

Agricultura de precisão e a IoT

Agora, você está trabalhando em uma empresa que presta serviços de automação para produtores rurais. Uma grande empresa do cultivo de mudas está interessada nos serviços de sua empresa de automação. Essa companhia utiliza grandes viveiros, isto é, áreas cobertas que visam proteger e acelerar

o crescimento das mudas, para o cultivo e desenvolvimento destas. A Figura 1.4 mostra um exemplo desse tipo de aplicação.

Figura 1.4 | Exemplo de um viveiro de mudas inteligente



Fonte: adaptada de Vieira (2017).

Essas estufas, conhecidas como casas de vegetação, são importantes no manejo dessas mudas. Isso ocorre pois esse tipo de estrutura possibilita um maior acompanhamento do crescimento das mudas, facilita o combate a pragas e controla o microclima gerado dentro da estufa. Sendo assim, a sua empresa foi contratada para realizar um levantamento inicial sobre quais sensores e atuadores devem ser empregados para assegurar um desenvolvimento saudável das mudas.

Os peritos em agricultura de precisão disseram que, para um manejo correto das mudas, é necessário que algumas variáveis sejam, minuciosamente, monitoradas e controladas. Essas variáveis são: umidade do solo, temperatura e umidade do ar, irradiação solar no interior da estrutura, concentração de CO₂ e níveis de água nos reservatórios. Você deve fornecer um levantamento inicial sobre quais sensores devem ser utilizados para garantir um monitoramento eficiente dessa casa de vegetação. Além disso, deve estar presente a indicação de quais atuadores serão necessários para construir essa aplicação em IoT.

Resolução da situação-problema

A Figura 1.5 apresenta um exemplo de combinação de sensores e atuadores. Para cada uma das variáveis a serem monitoradas, temos os seguintes sensores correspondentes:

Umidade do solo: sensor de capacidade variável, que utiliza a capacidade para medir a permissividade dielétrica do solo e, assim, estimar qual é a proporção de água existente.

Temperatura e umidade do ar: como exemplo temos o sensor DHT22, que apresenta tanto medidores de temperatura quanto de umidade atmosférica.

Irradiação solar: como exemplo temos o sensor LDR, que varia a resistência elétrica conforme a intensidade de luz recebida.

Concentração de CO₂: pode-se utilizar um sensor capaz de estimar a quantidade de CO₂ dentro da estufa, devendo este ser capaz de identificar pequenas variações na concentração de CO₂.

Níveis de água nos reservatórios: podem-se utilizar os sensores de nível como boias eletrônicas ou sensores de ultrassom para medir a altura da coluna de água nos reservatórios.

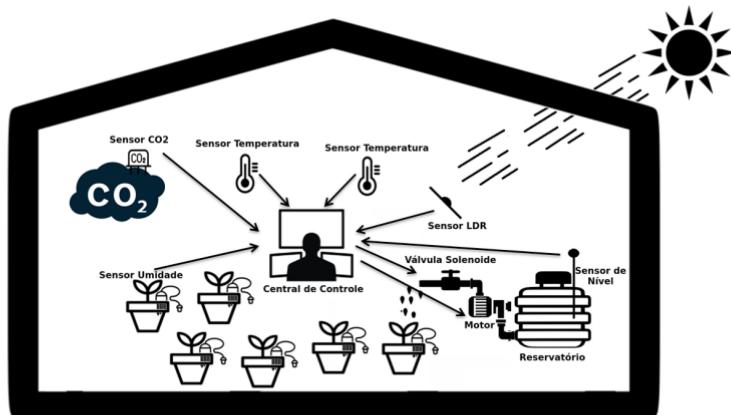
A seguir, conheça alguns exemplos de atuadores:

Motores: podem ser empregadas bombas hidráulicas para manter os níveis dos reservatórios constantes e controlar a umidade do solo e do ar. Quanto aos motores, estes podem ser empregados para abrir ou fechar persianas, a fim de controlar a quantidade de luminosidade incidente no interior da estufa.

Válvulas solenoides: controlam os períodos de irrigação das mudas.

Existem outros sensores e atuadores que podem ser utilizados. Além disso, vale acrescentar que é necessário sempre ouvir o seu cliente para que sejam alinhadas as expectativas e a realidade.

Figura 1.5 | Exemplo de possível solução para o problema



Fonte: elaborada pelo autor.

Faça valer a pena

1. Em 1999, Kevin Ashton foi o primeiro a empregar o termo Internet das Coisas. Ele utilizou esse termo para descrever um sistema no qual a internet estaria conectada com o mundo físico por meio de sensores ubíquos. Segundo o pesquisador, para que os dispositivos pudessem interagir entre si, por meio da internet, seria necessário que cada mensagem enviada encontrasse o destinatário desejado de maneira assertiva. Essa característica foi fundamental para o desenvolvimento da IoT e, como tal, foi incorporada em um dos seis elementos constituintes desse sistema. O RFID é um dos principais protocolos utilizados para alcançar a premissa proposta por Kevin Ashton (ASHTON *et al.*, 2009).

O elemento fundamental presente na fala de Kevin Ashton necessário para que uma mensagem encontre o seu destinatário final é o(a):

- a. Computação.
- b. Serviços.
- c. Semântica.
- d. Sensoriamento.
- e. Identificação.

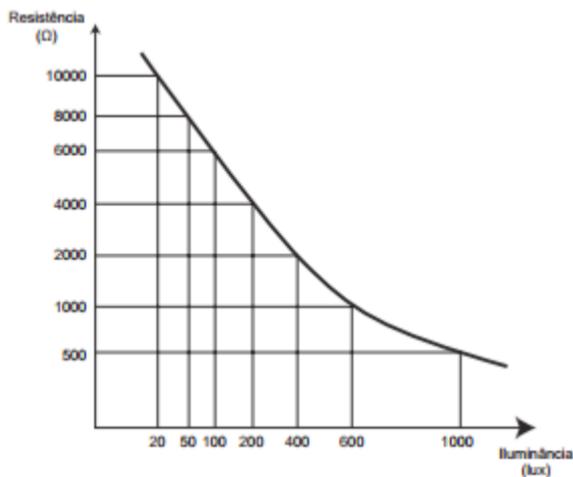
2. Dispositivos que compõem a Internet das Coisas precisam sobreviver em ambientes severos, realizando tarefas diversas. As redes formadas por esses dispositivos são, constantemente, acometidas com falhas nos nós e, portanto, os protocolos que trabalham com o roteamento da informação devem estar preparados para minimizar as perdas de sinal da rede. Todos esses problemas são gerados por características construtivas desses dispositivos. Essas características sempre devem ser levadas em consideração no momento em que estamos construindo uma aplicação em IoT. Saber minimizar as falhas é um preceito fundamental de qualquer protocolo para a IoT (VIEIRA, 2017).

Assinale alternativa que melhor explica as falhas constantes destes nós.

- a. Os dispositivos (nós) da rede IoT são muito caros para serem submetidos a condições severas.
- b. Todos os dispositivos são compostos por um processador, uma memória, uma bateria e um transmissor. Esses elementos são frágeis, fato este que confere funcionamento intermitente aos nós.
- c. Os dispositivos (nós) de uma rede apresentam baixa capacidade de processamento, fonte escassa de energia e estão sujeitos à interferência eletromagnética.

- d. Os nós da rede IoT não podem ser submetidos a elevadas vibrações e temperaturas, pois apresentam muitas partes móveis.
- e. Devido à extrema velocidade dos processadores, os nós em uma rede IoT sofrem com o aumento de temperatura.

3. A principal utilidade de uma função de transferência é realizar o mapeamento entre um valor de entrada e a saída. Devido a essa característica, utilizamos as funções de transferências dos sensores para escrever os programas de computador e realizar a automação dos processos (INSTRUMENTS, 1999). Imagine que temos um sensor LDR e que a função de transferência deste está representada pelo gráfico a seguir:



Fonte: Instruments (1999).

Por meio da análise desse gráfico, adotando-se apenas o período linear desse resistor (1000 e 10000 Ω), qual seria o valor aproximado de resistência para uma intensidade luminosa de 150lux:

- a. 5900 Ω .
- b. 1500 Ω .
- c. 3100 Ω .
- d. 7500 Ω .
- e. 4300 Ω .

Seção 2

Conceitos básicos de redes aplicados à IoT

Diálogo aberto

Na seção anterior, aprendemos os conceitos fundamentais sobre Internet das Coisas (IoT). Agora estamos preparados para darmos mais um passo em direção ao entendimento dos componentes e do que é necessário para solucionar problemas reais por meio da Internet das Coisas.

Questões relacionadas à segurança da informação e à privacidade dos dados são sempre assuntos primordiais para os sistemas modernos. Quando pensamos que na IoT existem bilhões de dispositivos conectados à internet e que, na maioria das vezes, esses dispositivos têm fonte escassa de recursos (baixo consumo de energia, reduzida capacidade de processamento e tamanho reduzido), é natural que a segurança e a privacidade sejam tratadas com atenção. Essas características de construção dos dispositivos também são fundamentais para a escolha do modo de comunicação e do modelo de rede a ser utilizada para a IoT (YANG, 2014).

Imagine que você ainda esteja trabalhando no projeto de construção da casa modelo de automação. Após o levantamento de quais os sensores e atuadores devem ser utilizados, é necessário indicar quais as questões de segurança da rede devem integrar esse projeto. Além disso, você deve sugerir o modelo de rede a ser utilizado na residência. Assim, os diretores desejam que você indique quais as questões de segurança e qual o modelo de rede são pertinentes ao desenvolvimento da casa modelo. Quais são as vulnerabilidades presentes na rede IoT dessa residência? E como podemos minimizá-las? Deve-se utilizar a comunicação cabeada ou sem fio? Qual topologia de rede seria mais interessante? Essas são algumas das perguntas que devem ser respondidas para serem adicionadas como elementos fundamentais em seu relatório de levantamento de requisitos.

No decorrer desta seção, vamos ajudá-lo a responder a essas perguntas. Concentraremos os nossos esforços nos conceitos que governam as redes de comunicação da Internet das Coisas e nas questões de segurança que permeiam o desenvolvimento da IoT. Em um primeiro momento, apresentaremos os conceitos fundamentais de uma rede de sensores/atuadores. Posteriormente, falaremos sobre como os dispositivos da IoT podem ser conectados. Nesse sentido, faremos uma comparação entre as conexões cabeadas e as sem fio. Em um terceiro momento, mostraremos as diferentes topologias de redes e as características que essas topologias transmitem para

IoT. Por último, serão apresentadas as questões relativas à segurança dos dispositivos IoT, além de como podemos minimizar esses riscos e, assim, construir aplicações mais seguras. Portanto, após a leitura desta seção, você será capaz de complementar o seu relatório de levantamento de requisitos e mostrar aos diretores que você está apto a assumir uma vaga na empresa.

Agora é a sua hora de trazer contribuições para a empresa e de mostrar que você tem as qualidades que os diretores estão buscando. Com grandes saberes vêm grandes responsabilidades, e você está a um passo de ser contratado. Vamos garantir a vaga?

Bons estudos!

Não pode faltar

Você já parou para pensar na quantidade de sensores e atuadores que estão presentes em nossas vidas? Imagine, por exemplo, em seu smartphone. Provavelmente, ele deve ter mais de uma dezena de sensores. Graças à presença massiva desses sensores, temos uma grande quantidade de dados sendo gerada a cada segundo. No ano de 2012, estimava-se que o montante de dados criados a cada dia, em todo o planeta, ultrapassava a casa dos 2,5 quintilhões de bytes (WALKER, 2015). Atualmente, esse montante de dados teve um aumento significativo.

Para que a informação gerada por esses dados possa ser utilizada por atuadores, de maneira automática, é necessário que existam redes de comunicação entre esses dispositivos. Além disso, é função das redes de comunicação possibilitar o compartilhamento de recursos entre os dispositivos e gerar redundância da informação (aumentar a confiabilidade dos dados), além de facilitar a configuração dos dispositivos e proteger a aplicação de possíveis ataques maliciosos (HANES *et al.*, 2017). Devido a todas essas funções, as redes de comunicação são fundamentais para a construção de aplicações em IoT.

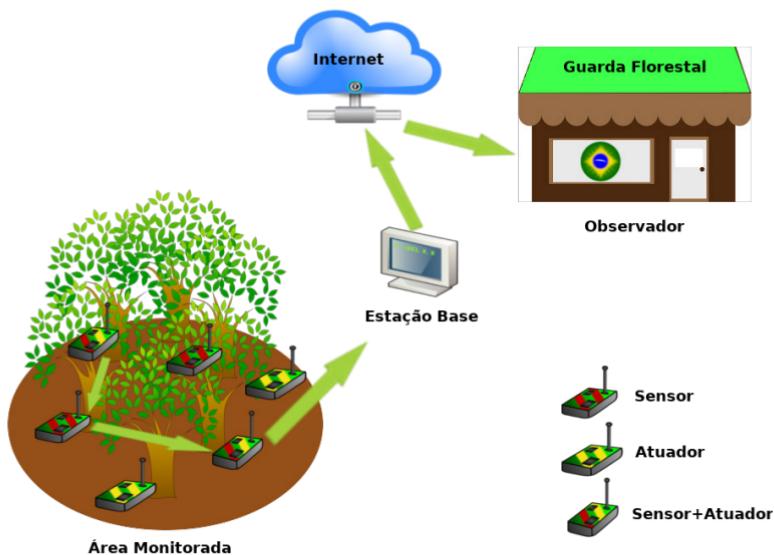
Ao equiparmos um dispositivo com sensores, atuadores, microprocessadores, memória, um componente de comunicação e uma fonte de energia, criamos um objeto inteligente (FORTINO, 2014). Esses objetos interagem entre si e com o ambiente físico ao seu redor por meio da internet como meio de comunicação, dessa forma, criamos a rede de dispositivos que constitui a Internet das Coisas.

As redes de sensores/atuadores são utilizadas para sentir (medir, se sensor) e interagir com o ambiente (se atuadores). Os sensores/atuadores dessa rede têm a capacidade de trocar informações e interagir de maneira cooperativa

em uma aplicação (HANES *et al.*, 2017). Nesse sentido, o principal desafio enfrentado pelas redes de sensores e atuadores é garantir que a comunicação e a cooperação entre os dispositivos da rede ocorram de maneira eficiente. Isso ocorre, pois esses dispositivos são diversos, heterogêneos e têm fonte limitada de recursos. Assim, garantir a eficiência na comunicação é a principal missão das redes de sensores/atuadores.

Os sensores/atuadores de uma rede são conhecidos como os nós da rede, pela estação base, pela aplicação em si (fenômeno de interesse) e pelo usuário dessa aplicação (observador) (YANG, 2014). Por exemplo, imagine uma aplicação em que se deseja monitorar e combater incêndios florestais de maneira automática. Os nós da rede podem ser compostos, por exemplo, de detectores de fumaça, termômetros, sensores de umidade, e de atuadores, como alarmes sonoros, bombas d'água, válvulas solenoides e drones. O fenômeno de interesse é composto da área de interesse que está monitorando e dos elementos que compõem essa área (vegetação e animais). O usuário ou observador pode ser, por exemplo, a unidade florestal responsável por essa área de mata. A Figura 1.6 ilustra os componentes principais presentes na rede de sensores/atuadores da aplicação descrita.

Figura 1.6 | Ilustração dos componentes principais de uma rede de sensores/atuadores



Fonte: elaborada pelo autor.

Existem duas formas distintas de comunicação entre os dispositivos de uma rede. A comunicação pode ser realizada utilizando cabos ou por meio do ar. Quando a troca de mensagens entre os sensores/atuadores ocorre através de condutores elétricos (cabos), temos o que chamamos de redes de sensores/atuadores cabeadas. Já quando a comunicação entre esses dispositivos ocorre através do ar, sem contato físico, denominamos de redes de sensores/atuadores sem fio. Para essa seção, vamos nos referir às redes de sensores/atuadores sem fio como RSSF. A escolha do meio para comunicação (cabo ou ar) influencia diretamente as características da rede. Assim, deve-se realizar uma comparação entre essas redes apresentando as vantagens e desvantagens de cada uma (KAUR, 2014).

As principais vantagens das redes cabeadas são as seguintes: confiabilidade, estabilidade, velocidade e segurança. Como os cabos condutores utilizados na rede cabeada apresentam blindagens eletromagnéticas, sofrem menor influência de outras conexões. Assim, as interferências podem ser minimizadas, resultando em uma maior estabilidade e confiabilidade no envio e no recebimento de mensagens. As redes cabeadas apresentam velocidades de comunicação superiores às apresentadas pelas RSSF. Como a comunicação em redes cabeadas é realizada diretamente entre o emissor e o receptor, a possibilidade de um agente malicioso interceptar essa mensagem é menor. Assim, a privacidade da informação é alcançada mais facilmente.

As desvantagens da utilização de redes cabeadas para a comunicação entre dispositivos são as seguintes: reduzida mobilidade, dificuldade de instalação e custo de manutenção. Como a comunicação é realizada por meio de conexão física entre os dispositivos, a mobilidade é restrita ao comprimento do cabo. A instalação dos dispositivos na rede apresenta uma maior complexidade, pois a infraestrutura necessária para conectar (cabos e portas) fisicamente todos os elementos da rede é maior quando comparada com as de conexões sem fio. Além disso, a manutenção e a atualização de uma rede cabeada é mais trabalhosa devido à necessidade de conectar todos os elementos através de cabos. A grande quantidade de cabos pode inviabilizar algumas aplicações.

Imagine ter de utilizar o seu notebook em sua casa e precisar trafegar pelos cômodos carregando metros de fios. Assim, para uma melhor mobilidade é mais adequada a utilização de redes sem fio.

A comunicação sem fio tem como principais vantagens, além da maior mobilidade, o menor tempo de instalação, escalabilidade e conectividade (KAUR, 2014). O tempo de implementação das RSSF é, geralmente, inferior ao das redes cabeadas, pois, além de não precisar realizar o cabeamento, a

infraestrutura existente pode ser utilizada. Por exemplo, não são necessárias obras de alvenaria, furos em paredes ou adaptações extras para permitir a conexão dos dispositivos da rede. Uma vez que não são necessários cabos, a restrição da mobilidade é imposta apenas pela área de cobertura da rede. Como não precisamos adicionar cabos a cada novo elemento integrado à rede, dimensionar a quantidade de nós é menos custoso. Assim, adicionar ou remover dispositivos dessa rede é mais rápido e fácil. Além disso, os usuários poderão se conectar e desconectar de forma mais rápida, devido a inexistência de cabos.

As redes sem fio, assim como as cabeadas, também apresentam desvantagens como a maior vulnerabilidade a ataques, menor velocidade de transmissão e menor confiabilidade e estabilidade da rede (KAUR, 2014). Como as RSSF são visíveis a outras redes, elas são mais vulneráveis a ataques. Além disso, como as mensagens são enviadas para todas as direções, é mais fácil para agentes maliciosos interceptarem essas informações trocas. Como, normalmente, as mensagens são trocadas através de ondas de rádio, a velocidade das RSSF é influenciada pela frequência e pela performance do rádio transmissor. Logo, as velocidades alcançadas são inferiores às obtidas pelas redes cabeadas. As RSSF não dispõem dos artifícios de blindagem durante todo o percurso da mensagem, entre a origem e o destino, então estão sujeitas a interferências. Essas interferências podem provocar degradação, modificação e até perda do conteúdo das mensagens trocadas. Assim, os protocolos devem ser robustos para minimizar os efeitos das interferências.

Como vimos, cada um dos dois tipos de redes apresenta vantagens e desvantagens. Você já deve ter percebido que a escolha entre um e outro deve ser realizada por meio da avaliação dos objetivos e recursos disponíveis para cada aplicação (KAUR, 2014). Quando trabalhamos com IoT geralmente preferimos utilizar as RSSF, devido à quantidade de dispositivos existentes e aos locais onde eles são instalados (monitoramento de florestas, vida marinha e vulcões).

Refita

A respeito das vantagens e desvantagens de cada um dos tipos de redes de sensores/atuadores (cabeadas e sem fio), foi dito que a confiabilidade das RSSF é inferior à cabeada. A justificativa apresentada foi de que não é possível garantir a blindagem contra interferência.

Você acha que é impossível realizar uma blindagem eletromagnética durante a transmissão sem fio? Devido às características dos objetos

inteligentes da IoT, como podemos reduzir as interferências através da comunicação sem fio?

A Internet das Coisas, assim como as redes de computadores, também apresenta a divisão em camadas. A divisão de uma rede em camadas tem como principal objetivo reduzir a complexidade do projeto da rede por meio do encapsulamento de dados (TANENBAUM, 2003). Existem três modelos sugeridos de arquitetura para a IoT, entre eles estão os modelos em três camadas (SANTOS *et al.*, 2016), em cinco camadas (KHAN *et al.*, 2012) e em sete camadas (HANES *et al.*, 2017). A Figura 1.7 apresenta uma representação dos modelos das três arquiteturas. Nossa foco será a arquitetura de cinco camadas.

Na arquitetura em cinco camadas, a IoT pode ser dividida em camada de percepção, camada de rede, camada de *middleware*, camada de aplicação e camada de negócio (SETHI; SARANGI, 2017). A camada de percepção ou camada dos dispositivos corresponde à primeira camada de uma rede IoT. É nela que encontramos os elementos responsáveis por sentir o ambiente e interagir com ele (sensores e atuadores). A camada de rede ou de transmissão é responsável por transmitir os dados, de maneira segura, para os sistemas de processamento da informação. Essa transferência pode ocorrer através de cabos ou por meio da comunicação sem fio. A camada *middleware* é responsável por assegurar a comunicação efetiva entre os diferentes dispositivos que compõem a IoT. A IoT é composta de milhões de dispositivos dos mais variados fabricantes, que disponibilizam diferentes serviços de conexão. Essa camada é responsável por gerenciar esses serviços disponibilizados pelos dispositivos e por garantir uma comunicação eficiente. Também é ela que recebe a informação da camada de rede e armazena os dados. A camada de aplicação é responsável por realizar o gerenciamento global da aplicação. É nessa camada que são implementadas as aplicações da IoT. Ela fornece uma interface para que os aplicativos que estiverem rodando nos dispositivos possam acessar a rede de comunicação. A camada de negócio é responsável por gerar “valor” para a aplicação. É aqui que são implementados os modelos de negócio, são gerados os gráficos, fluxogramas e as análises de dados. É a partir da análise realizada nessa camada que são gerados os subsídios utilizados pelos administradores da aplicação para tomarem decisões.

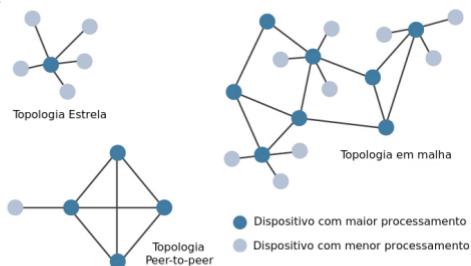
Figura 1.7 | Comparação entre diferentes modelos de arquiteturas sugeridas para IoT: a. modelo em três camadas, b. modelo em cinco camadas e c. modelo em sete camadas



Fonte: adaptada de Sethi e Sarangi (2017).

Outra característica que influencia o desempenho de uma rede de comunicação é o tipo de topologia adotada. A topologia indica como os dispositivos da rede serão conectados entre si e o modo como é realizada a troca de informação entre eles. Alguns requisitos a serem analisados para a escolha adequada da topologia são os seguintes: facilidade de manutenção, confiabilidade, escalabilidade e consumo de energia (HANES *et al.*, 2017). Apresentaremos os três tipos de topologia mais utilizados em IoT: a topologia estrela, a *peer-to-peer* (P2P) e a em malha. Essas três topologias estão retratadas na Figura 1.8.

Figura 1.8 | Principais topologias empregadas em redes de comunicação



Fonte: adaptada de Hanes *et al.* (2017).

Na topologia estrela, existe uma estação base ou controlador que é utilizado como centralizador da informação. É a estação base que permite a comunicação em cada um dos nós localizados nas extremidades da rede. A maior desvantagem dessa topologia é o fato de que falhas na estação base podem ocasionar um colapso da rede (YANG, 2014).

A topologia *peer-to-peer* permite que qualquer dispositivo da rede consiga estabelecer comunicação com qualquer outro dispositivo que esteja dentro da área de cobertura do sinal. A comunicação é realizada aos pares. Devido à comunicação apenas entre dois dispositivos, a topologia P2P não é muito utilizada em redes IoT (YANG, 2014).

Em topologias do tipo em malha cada um dos nós da rede é conectado a múltiplos outros nós. As mensagens são trocadas entre os nós até que o destinatário seja alcançado. Como não existe apenas um elemento centralizador, as redes em malha também apresentam maior confiabilidade. Essa topologia é a mais utilizada em aplicações IoT, pois possibilita uma maior confiabilidade, escalabilidade e facilidade de manutenção à aplicação. Todos os protocolos mais modernos, como o 6LoWPAN (SHELBY e BORMANN, 2011), o ZigBee (FARAHANI, 2011), o Bluetooth Low Energy (BLE) (NIEMINEN *et al.*, 2015), e demais protocolos do padrão IEEE 802.15.4, suportam esse tipo de topologia.

Pesquise mais

Como leitura complementar, sugerimos o artigo a seguir, em que são apresentados vários conceitos, aplicações e uma discussão sobre os diferentes tipos de redes de comunicação.

LOUREIRO, A. A. F. *et al.* Redes de sensores sem fio. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES, 21., 2003, Natal. **Anais** [...]. Natal: SBRC, 2003. p. 179-226.

Como foi possível notar, não existe uma topologia que seja a ideal para todas as aplicações. A escolha de qual modelo utilizar deve ser realizada após analisar o ambiente em que será desenvolvida a aplicação e quais os objetivos devem ser alcançados pela solução em IoT proposta. Quando a adoção de uma única topologia não é satisfatória, pode-se utilizar a combinação dessas topologias para que sejam alcançados os resultados esperados.

Segurança em IoT

Como já vimos, a IoT é composta de bilhões de dispositivos, provenientes de milhares de fabricantes diferentes, realizando as mais variadas funções, tudo isso através de dispositivos com recursos limitados. Em

qualquer tipo de rede (computadores, sensores e celulares) a segurança da informação é primordial, mas, devido às suas características peculiares, para a IoT, devem ser observados alguns pontos específicos a fim de garantir a segurança da informação. Assim, para alcançarmos a segurança de aplicações em IoT devemos garantir a privacidade, a confidencialidade, a integridade da infraestrutura da rede, dos dados e dos dispositivos (nós) que compõem a IoT (HASSAN, 2019).

Conforme já aprendemos, podemos dividir a IoT em dois grandes níveis: nível de dispositivo e nível de comunicação. No nível dos dispositivos, são encontrados os sensores/atuadores, enquanto no nível de comunicação ocorre a troca de mensagens entre os dispositivos, ou seja, é nesse nível que é encontrada a rede de comunicação.

Os hardwares (sensores e atuadores) são encontrados no nível dos dispositivos. Para que eles possam desempenhar essas funções (monitoramento e interação com o ambiente) é necessário que sejam desenvolvidos softwares. Nesse sentido, os softwares é que indicam como o hardware deve executar as tarefas de monitoramento, atuação e comunicação (SUN, 1999). A segurança de hardware e de software nesse nível é crucial para preservar a privacidade dos usuários, assegurar a confiabilidade dos dados e certificar a segurança da infraestrutura (HERSENT; BOSWARTHICK; ELLOUMI, 2011).

Uma das características do nível dos dispositivos (nível 1) é realizar e manter a identificação única dos dispositivos. Quando algum dispositivo malicioso tem acesso à identificação única de um nó da rede, esse agente malicioso pode ingressar na rede de comunicação. Ao ter acesso à rede de comunicação, esse agente terá acesso aos dados trocados pela rede. Além disso, esse dispositivo conseguirá enviar mensagens de maneira indiscriminada para os demais dispositivos, o que pode levar ao colapso da aplicação. Para proporcionar segurança à identificação dos dispositivos é necessário que o sistema operacional (S.O), a inicialização do sistema (*boot*) e o armazenamento sejam operações seguras. Para conseguir uma identificação segura é necessário um *boot* confiável (HASSAN, 2019). Um *boot* confiável significa que quando o hardware for ligado, o *firmware* verificará se as assinaturas de cada parte dos *drivers* e softwares são confiáveis. Ou seja, o software executado não apresentará ameaças ao sistema. Para um armazenamento seguro das chaves de identificação, pode ser utilizada a criptografia e os meios de autenticação. Essas estratégias visam dificultar o acesso à chave de identificação do dispositivo. Assim, a identificação poderá ser confirmada e apenas os hardwares corretamente identificados poderão fazer parte da rede.

Assimile

Os *firmwares* são instruções que são gravadas diretamente na memória não volátil do hardware. Ou seja, essas instruções não são perdidas quando ocorre uma falha de energia. O *firmware* pode ser armazenado na PROM ou EPROM. É ele o responsável por manter as funções básicas de funcionamento do sistema.

Um exemplo, é a BIOS (Basic Input/Output System) do seu computador. Esse é o primeiro programa a ser executado, é ele que carrega os *drivers* dos hardwares (placa de rede, placa vídeo e disco rígido) e o sistema operacional (SUN, 1999).

No segundo nível (nível de comunicação), encontra-se a rede de comunicação. É nesse nível que os protocolos de comunicação devem garantir a privacidade da informação e assegurar o correto funcionamento da rede. Além disso, é através desse nível que os dispositivos realizam a comunicação, por meio dos *gateways*, com os elementos presentes na computação em nuvem. Desse modo, é necessário prover a segurança tanto dentro da rede local quanto na comunicação com a nuvem de dados.

Para a segurança da informação em redes de comunicação, como as redes de computadores e redes industriais, existem vários sistemas e protocolos para permitir a troca segura de mensagens entre os nós dessas redes. Entretanto, como você já sabe, os nós em redes IoT apresentam disponibilidade escassa de recursos. Logo, os protocolos e sistemas de segurança utilizados nas redes convencionais não podem ser empregados diretamente nas redes IoT. Por exemplo, para a comunicação por meio da internet, o protocolo IPv6 não pode ser utilizado diretamente nos nós da rede IoT, uma vez que esse protocolo, dentre outras questões, não é otimizado para o consumo de energia (SHELBY; BORMANN, 2011).

A fim de promover uma comunicação segura tanto dentro da rede local quanto entre os dispositivos e nuvem, vários protocolos foram desenvolvidos. Protocolos como o 6LoWPAN, ZigBee e MQTT contam com métodos de autenticação, criptografia, prevenção de invasão e o roteamento seguro da informação (AL-FUQAHÁ, 2015). Essas medidas visam evitar que invasores possam ingressar na rede, capturar as mensagens e prejudicar o funcionamento normal da aplicação. Assim, esses protocolos empregam estratégias já existentes, mas com enfoque nas peculiaridades presentes nas aplicações IoT.

Para as plataformas de computação em nuvem, também é necessário que estratégias de prevenção e combate a agentes maliciosos sejam adotadas. As plataformas de computação em nuvem de grandes empresas utilizam criptografia, autenticação e gerenciamento de usuários para garantir o

armazenamento, o processamento e a visualização segura dos dados enviados (HASSAN, 2019). Portanto, como você deve ter percebido, as questões de segurança ainda representam um grande desafio para a IoT, uma vez que as características dos dispositivos e a construção dessa rede fazem com que ela seja mais vulnerável a ataques (ZHANG *et al.*, 2014).

Exemplificando

Como vimos, a segurança para a IoT é fundamental para o sucesso da aplicação. Quando a segurança em cada um dos níveis não é alcançada, sérios problemas podem ocorrer.

Um exemplo de falha de segurança bastante comentado é o caso do veículo da Tesla. Em 2016, pesquisadores chineses, conseguiram invadir os sistemas de segurança do Tesla Model S. A uma distância de aproximadamente 20 km, esses pesquisadores obtiveram o controle de vários sistemas vitais para a operação do veículo. Eles conseguiram controlar os sistemas de freios, fechamento das portas, computador de bordo e outros sistemas eletrônicos. Ao demonstrar a vulnerabilidade desse veículo a ataques maliciosos, os pesquisadores comprovam que a segurança em IoT deve ter um papel central no desenvolvimento de qualquer aplicação IoT (SOLON, 2016).

Nesta seção, apresentamos os conceitos principais sobre a estrutura das redes IoT, discutimos as vantagens e desvantagens de cada tipo de rede e falamos sobre segurança. Após estudar esse conteúdo, você já é capaz de compreender melhor a estrutura de uma aplicação IoT. Agora você está preparado para definir qual o tipo de rede é mais adequado para uma determinada aplicação.

Sem medo de errar

Agora que você já realizou o levantamento inicial dos sensores e atuadores a serem utilizados na casa modelo em automação, chegou a hora de mostrar aos diretores que você também sabe qual é a melhor configuração para a rede e que a segurança é fundamental para a aplicação. Para isso, você deve indicar qual tipo de rede (cabeadas ou sem fio) e topologia deve ser utilizada, destacando suas vantagens e limitações, além de apresentar quais as questões de segurança devem ser observadas na construção da rede. Assim, o relatório apresentado aos diretores estará mais completo e o projeto de automação, mais próximo de alcançar os objetivos.

Como estudamos nesta seção, existem dois tipos de redes: cabeadas e sem fio. Cada uma dessas configurações apresenta vantagens e desvantagens. Portanto, é necessário observar quais são as características da aplicação, pois isso é fundamental para que seja realizada uma escolha adequada do tipo de rede a ser utilizado. Também estudamos as principais topologias empregadas na construção da IoT, pois, assim como na escolha do tipo de rede, a topologia a ser escolhida deve satisfazer os requisitos da aplicação. Outro ponto que abordamos nas discussões desta seção foi relativo às questões que envolvem a segurança da IoT. Aprendemos quais são as principais ameaças à segurança e quais ações são realizadas a fim de mitigá-las. Esses pontos são de extrema importância para que sejam desenvolvidas redes IoT robustas e eficientes. Desse modo, você tem todo o conhecimento necessário para completar o relatório e convencer os diretores de todo o seu potencial.

Considerando essas informações, o tipo de conexão escolhida é a sem fio. Como estamos trabalhando com a automação de uma residência, que já está construída, não desejamos investir em obras de alvenaria. Não desejamos quebrar paredes, fazer furos ou ter de alterar as instalações elétricas já existentes. Para a utilização de uma rede cabeada de comunicação, é necessária uma adequação do espaço em que a aplicação de automação será construída. Esse é o primeiro ponto a favor da escolha das redes sem fio.

Como segundo ponto, temos que, em redes de sensores/atuadores sem fio, a instalação e a configuração dos dispositivos são questões mais fáceis de serem resolvidas. Isso ocorre, pois a escalabilidade é um fator decisivo para essa aplicação em domótica, uma vez que, normalmente, os clientes não realizam a automação de toda a residência em apenas uma etapa. Logo, utilizar a comunicação sem fio possibilita a escalabilidade da automação dessa residência.

Outro ponto a favor da escolha das redes sem fio é a mobilidade proporcionada por esse tipo de rede. Essa questão é fundamental, pois alguns objetos inteligentes, sensores e atuadores necessitam ter mobilidade para facilitar a vida dos moradores da residência. Por exemplo, os moradores podem desejar transportar os equipamentos eletrônicos, como notebook, carregadores de celular ou ventiladores, pelos cômodos da casa, sem abdicar de contar com a comodidade de desligamento automático após deixarem de utilizá-los. Essas são apenas algumas das características das redes de sensores/atuadores sem fio que favorecem a sua utilização nessa aplicação em domótica. O Quadro 1.3 apresenta o resumo das qualidades das redes sem fio, necessárias para a resolução da situação-problema.

Quadro 1.3 | Resumo das qualidades da rede sem fio

Tipo de rede escolhida	Qualidades necessárias
Redes sem fio	Não necessita de obras de alvenaria.
	Maior facilidade de instalação.
	Maior escalabilidade.
	Maior mobilidade.

Fonte: elaborado pelo autor.

A topologia de rede escolhida é a topologia em malha. Ela apresenta a vantagem de conseguir lidar com uma grande quantidade de mensagens trocadas. Como todos os cômodos da casa modelo serão automatizados, com a utilização de um grande número de sensores/atuadores, essa aplicação deve gerar uma grande quantidade de mensagens que serão trocadas entre todos os nós da rede. Além disso, os protocolos mais utilizados em aplicações IoT têm suporte a essa topologia. Em uma rede em malha, falhas em um dos nós não provocam um colapso em toda a rede, logo, essa é uma característica interessante para a construção da casa modelo, visto que queremos construir uma rede confiável. A questão da escalabilidade da rede também é outro fator que influencia a escolha pela topologia em malha, já que nessa topologia a introdução de novos dispositivos não afeta a troca de mensagens através da rede.

Os pontos relativos à segurança, observados para a automação da residência, são os mesmos de qualquer outra aplicação IoT. Entretanto, alguns pontos relativos à detecção de invasores são de fundamental importância. Como serão trocados dados relativos à segurança residencial, por exemplo, imagens das câmeras de vigilância, esses dados não podem ser obtidos por agentes maliciosos, pois, em mãos erradas, representam uma ameaça à segurança dos moradores.

Portanto, devem ser utilizados sistemas de criptografia, autenticação, detecção e expulsão de invasores. Para o nível dos dispositivos, devem ser empregados sistemas de proteção para os hardwares. Esses sistemas precisam garantir a identificação segura dos dispositivos, por exemplo, com a utilização de boot e armazenamento seguros. No segundo nível, as mensagens trocadas devem ser recebidas apenas pelos integrantes da aplicação. Para manter essa comunicação segura, é necessário que todos os protocolos utilizados implementem algoritmos de criptografia e autenticação. Com a adoção dessas medidas, a automação dessa residência estará dentro dos principais parâmetros de segurança existentes na IoT.

Veículo autônomo

Após você aprender sobre domótica, você iniciou a busca por conhecimento de Internet das Coisas aplicado em outras áreas. Várias empresas estão interessadas em seus serviços, pois ficaram encantadas com todos os seus trabalhos realizados. Agora, estão disputando esse profissional talentoso que você se tornou. Após analisar várias propostas, você decidiu ingressar no ramo automotivo.

Uma grande empresa multinacional escolheu você para ser o gerente de desenvolvimento do novo carro autônomo. Esse veículo deve apresentar todas as tecnologias mais modernas existentes. Em um primeiro momento, foram levantados os requisitos para a construção desse veículo e, com isso, foram selecionados os sensores e atuadores para esse projeto. O carro deve ser capaz de garantir que todos os passageiros consigam chegar em segurança a um destino escolhido de uma viagem. Além disso, devem ser observadas todas as leis de trânsito e a segurança dos demais usuários das vias terrestres.

Você não precisa se preocupar com quais sensores/atuadores precisam ser utilizados, pois essa etapa já foi realizada. Entretanto, os diretores esperam que você entregue um relatório contendo qual o tipo de rede a ser empregado (cabeada ou sem fio), qual a topologia de rede mais indicada para essa aplicação (estrela, em malha ou *peer-to-peer*) e quais questões de segurança precisam ser avaliadas. Os diretores ainda precisam que você apresente argumentos que comprovem a escolha por cada um dos parâmetros de construção selecionados.

Resolução da situação-problema

A conexão indicada entre os dispositivos é a cabeada. Pois, como a velocidade, a confiabilidade e a segurança são os principais e essenciais requisitos a serem cumpridos para que um veículo autônomo possa realizar todas as funções necessárias a uma condução segura (avaliar a condição das vias, identificar faixas, pedestres, veículos e placas de trânsito). Assim, o tipo de conexão que fornece esses elementos é a cabeada.

A topologia mista é a mais indicada para essa aplicação, pois, devido à complexidade do sistema, o mais adequado é a junção de várias topologias. Dessa forma, poderá ser utilizada a topologia mais adequada para a especificidade. Por exemplo, para uma comunicação direta entre os dispositivos

sensores e a central de processamento ocorrer de maneira mais rápida, pode-se utilizar a topologia *peer-to-peer* ou a topologia estrela. Já para o armazenamento ou para a atualização dos dados empregados pelos algoritmos de aprendizado de máquina, pode-se utilizar a topologia em malha, uma vez que é necessário lidar com um grande tráfego de informação e a velocidade não é o maior gargalo. Assim, uma solução que integra as vantagens de cada uma das topologias é a mais indicada para essa aplicação.

Como relatado nesta seção, a segurança deve ser observada em todos os dois níveis da aplicação. Os hardwares devem ser protegidos contra a instalação de programas maliciosos e os softwares devem apresentar sistemas de autenticação e criptografia. A rede de comunicação deve ser construída por meio de protocolos que garantam a criptografia e a identificação dos dispositivos que compõem a rede. Agentes maliciosos devem ser prontamente identificados e eliminados do sistema. Como visto, a segurança de veículos autônomos é de fundamental importância para desenvolvimento dessa aplicação de IoT.

Faça valer a pena

1. Um dos protocolos mais conhecidos e empregados pela IoT é o 6LoWPAN. Ele é empregado para possibilitar a utilização do protocolo IPv6 em dispositivos com disponibilidade reduzida de recursos e que integram redes com baixas taxas de transferências de mensagens, grande quantidade de perdas de pacotes e baixo consumo de energia. Essas redes são conhecidas como Low-Power and Lossy Network (LLN) (SHELBY; BORMANN, 2011).

A camada ou subcamada da IoT que pertence ao protocolo 6LoWPAN é a:

- a. Camada de rede.
- b. Camada de aplicação.
- c. Camada de enlace.
- d. Subcamada de adaptação.
- e. Camada IEEE 802.15.4.

2. A segurança da informação não é uma prioridade apenas para a IoT. Em redes de computadores e industriais ela é crucial para a privacidade e desempenho da aplicação. Para a IoT, essa questão é ainda mais dramática, pois existem bilhões de dispositivos, de diferentes fabricantes e com características próprias, conectados à internet (ZHANG *et al.*, 2014). Assim, garantir que a invasão de agentes maliciosos seja cercada torna-se uma tarefa mais trabalhosa. Deve-se assegurar a integridade e

a privacidade dos dados em todos os níveis da aplicação (ZHANG *et al.*, 2014). Além disso, deve-se garantir que apenas os participantes da rede tenham acesso às mensagens trocadas (ZHANG *et al.*, 2014).

A estratégia utilizada para garantir que a identidade do nó corresponda a um elemento conhecido pela rede é:

- a. A confiabilidade.
- b. A decodificação.
- c. O salto de canal.
- d. A autenticação de dispositivos.
- e. A blindagem eletromagnética.

3. O padrão IEEE 802.15.4 define as especificações de como deve ser o acesso ao meio físico (ambiente) da aplicação e de como os dados obtidos no ambiente devem ser acessados pelas camadas superiores (ADAMS, 2006). Além disso, o padrão IEEE 802.15.4 apresenta as definições de *Full Function Device* (FFD) e *Reduced Function Device* (RFD). O FFD é um dispositivo que pode ser utilizado como roteador ou coordenador da rede. Eles apresentam mais funcionalidades do que os RFD. O papel dos RFD é o de realizar tarefas simples, por exemplo, acender ou apagar uma lâmpada. Os FFD podem se comunicar tanto com outros FFD quanto com os RFD. Já os RFD só podem se comunicar com os FFD.

As camadas ou subcamadas que implementam o protocolo IEEE 802.15.4 são:

- a. Enlace e adaptação.
- b. Aplicação e transporte.
- c. Enlace e transporte.
- d. Física e adaptação.
- e. Física e enlace.

Seção 3

Protocolos de comunicação aplicados à IoT

Diálogo aberto

Você já parou para pensar que todos os dias utiliza diferentes tipos de tecnologias para se comunicar? Por exemplo, em uma ligação telefônica, você utiliza as células de telefonia para se conectar ao roteador sem fio de sua residência. Quando você não está em sua residência, é por meio do 4G que acessa as redes sociais e no momento em que você escuta as músicas de seu smartphone na multimídia do seu carro, está utilizando o Bluetooth. Esses são alguns exemplos de protocolos que empregamos em nossas tarefas diárias.

Os protocolos de comunicação são fundamentais para qualquer sistema que processa, armazena e troca dados ou informações. A escolha dos protocolos é uma tarefa difícil devido à grande quantidade de dispositivos e a diversidade de fabricantes. Isso ocorre, porque a IoT é composta por uma grande quantidade de dispositivos com características próprias. Assim, esta é uma rede naturalmente composta por elementos com características heterogêneas que devem se comunicar de maneira eficiente.

Assim, chegamos, agora, à terceira e última etapa para a elaboração do relatório sobre quais requisitos são necessários para a construção da casa modelo. Essa residência deve conter os protocolos mais seguros e mais eficientes disponíveis no momento, pois ela será utilizada como vitrine para atrair novos clientes.

Nesta seção, serão abordados os principais protocolos existentes e como eles possibilitam o desenvolvimento de aplicações IoT. Além disso, serão estudadas as características de cada um desses protocolos, bem como as diferenças existentes entre esses protocolos e os desenvolvidos para as redes de computadores convencionais. Para isso, vamos apresentar a arquitetura da Internet das Coisas e quais as camadas que compõem essa rede e como cada uma dessas camadas está relacionada com as demais.

Dessa forma, a terceira atividade que os diretores da empresa desejam no seu relatório é o levantamento de quais protocolos devem ser utilizados para a integração entre os sensores, atuadores e a internet. Portanto, você deverá refletir e pensar quais os protocolos devem estar presentes em cada uma das camadas da IoT. A escolha adequada é fundamental, uma vez que garantem a interoperabilidade entre todos os componentes da rede. O ponto central da terceira etapa desse relatório, é refletir sobre como os protocolos ajudam a mitigar as limitações impostas pela IoT. Nesse sentido, você deverá

ter um amplo conhecimento sobre a arquitetura da Internet das Coisas e como as camadas dessa arquitetura estão relacionadas. Ademais, é necessário conhecer os principais protocolos presentes em cada uma dessas camadas para que seja possível compará-los e, assim, indicar qual é o mais apropriado para cada situação. Esses conhecimentos são fundamentais para a construção de uma aplicação robusta e eficiente.

Assim, você será capaz de propor soluções que integrem de forma eficiente os diferentes tipos de dispositivos. Com isso, o seu trabalho contribuirá para o desenvolvimento da empresa e com sua melhor formação profissional. Os diretores confiam em você. Está preparado para crescer profissionalmente?

Não pode faltar

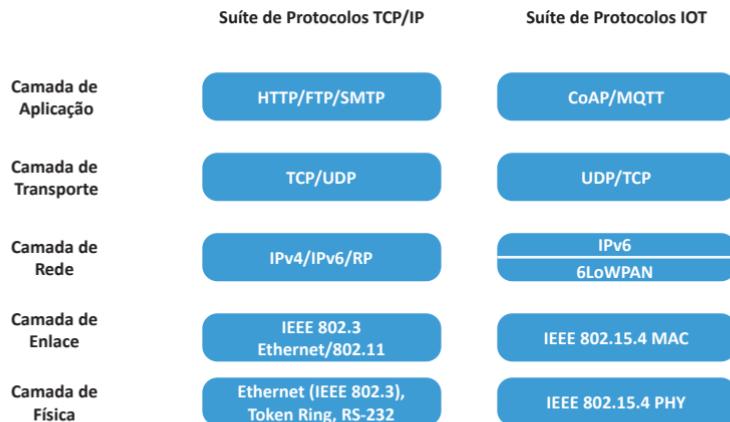
Em 1969 foi criada, nos Estados Unidos, a ARPANET (do inglês, *Advanced Research Projects Agency Network*). Essa rede foi desenvolvida com o propósito de interligar os laboratórios de pesquisas e, assim, permitir que os pesquisadores compartilhassem ideias e os avanços alcançados em suas pesquisas. Anos depois, após uma série de melhorias, essa rede entre pesquisadores passou a interligar vários pontos do mundo. Com os avanços tecnológicos e científicos foi criada a internet que possibilitou a comunicação entre computadores e pessoas espalhadas ao redor do planeta. Com o desenvolvimento da Internet os sistemas embarcados, também evoluíram e deram origem a sistemas microcontrolados cada vez menores e com maior capacidade de processamento e comunicação. Com a criação de protocolos mais robustos e focados para a utilização nesses sistemas embarcados, passamos a ter um cenário adequado para o surgimento da Internet das Coisas.

A Internet das Coisas, assim como as redes de computadores e outros sistemas de comunicação, precisam de protocolos para que a troca de mensagens entre todos os elementos que compõem a rede possa ocorrer de maneira eficiente. São os protocolos que definem um conjunto de regras que devem ser seguidas para que um dispositivo possa se comunicar com outro (BONAVVENTURE, 2011).

A IoT pode ser dividida em 5 diferentes camadas. Essas camadas são similares às encontradas na pilha de protocolos TCP/IP (*Transmission Control Protocol/Internet Protocol*), mas como veremos no decorrer desta seção, existem diferenças fundamentais que tornaram possível o surgimento da IoT.

A Figura 1.9 apresenta a pilha de protocolos TCP/IP em comparação com a arquitetura da IoT. Como pode ser visto, apesar de possuírem a mesma quantidade de camadas e a existência de alguns protocolos em comum. Essas arquiteturas são utilizadas em dispositivos com diferentes características. Sendo assim, mesmo compartilhando características similares, devem existir modificações para que ambos os sistemas possam se comunicar de maneira eficaz.

Figura 1.9 | Comparação entre as suítes de alguns protocolos da pilha TCP/IP e IoT



Fonte: adaptada de White, Fish e Pooch (2017, [s.p.]).

A característica que possibilita a disseminação da IoT em quase todas as áreas de conhecimento é o fato dela conseguir utilizar o protocolo IPv6 em seus dispositivos com limitada fonte de recursos. Isso se deve às adaptações realizadas pelos protocolos em cada uma das camadas presentes na Figura 1.9, em especial a utilização de uma adaptação presente entre a camada de enlace e a camada de rede.

Os protocolos desenvolvidos para a camada física e de enlace, precisam garantir uma comunicação eficiente entre os dispositivos da IoT. Os protocolos mais conhecidos dessas camadas que lidam, simultaneamente, com todas essas características são os seguintes: IEEE 802.15.4 (GUTIERREZ; CALLAWAY; BARRETT, 2004), o IEEE 802.15.4E (KURUNATHAN *et al.*, 2018), o Low Power Wi-Fi (PARK, 2015) e o LoRaWAN (SORNIN *et al.*, 2015).

A fim de criar ou adaptar os protocolos já existentes a fim de serem utilizados de maneira eficaz para a comunicação entre os dispositivos da IoT, foi lançado, em maio de 2003, o padrão IEEE 802.15.4 (GUTIERREZ; CALLAWAY; BARRETT, 2004). Esse padrão apresenta as diretrizes a serem

implementadas para desenvolver protocolos eficientes para dispositivos que utilizam baixas taxas de transferência de dados, disponibilidade escassa de recursos e limitada capacidade de processamento (YANG *et al.*, 2014). Assim, esse padrão visa permitir a construção de protocolos semelhantes aos existentes na comunicação sem fio, por exemplo, Wi-Fi e Bluetooth, adotando as particularidades dos dispositivos da IoT. O padrão IEEE 802.15.4 define as especificações sobre como deve ser o acesso ao meio físico (ambiente) da aplicação e como os dados obtidos no ambiente devem ser acessados pelas camadas superiores. Esse padrão implementa as diretrizes de utilização para as camadas físicas e de enlace por meio, respectivamente, dos padrões IEEE 802.15.4 PHY e IEEE 802.15.4 MAC.

Em 2007, a IEEE propôs algumas mudanças no protocolo IEEE 802.15.4 a fim de adequar esse protocolo para a utilização em ambiente industrial. Assim, surgiu o IEEE 802.15.4E, que contém as mesmas características de construção do IEEE 802.15.4, mas com melhorias na segurança, na confiabilidade e na redução do consumo de energia (VASSEUR *et al.*, 2010). Entretanto, o suporte a esse protocolo não está disponível na maioria dos dispositivos IoT.

O Wi-Fi (abreviação de *wireless fidelity*) que conhecemos e utilizamos diariamente para a comunicação sem fio entre computadores, smartphones e tablets, não foi projetado para ser utilizado em dispositivos com restrições de energia. O Wi-Fi (padrão IEEE 802.15.11) é otimizado para gerar respostas rápidas com alta taxa de transmissão e baixa latência. Essas características fazem com que o consumo de energia seja mais elevado (VASSEUR *et al.*, 2010). Já o Low Power Wi-Fi foi concebido para gerar um baixo consumo de energia. O Low Power Wi-Fi consegue reduzir o consumo de energia, pois, diferentemente do Wi-Fi convencional, não permanece em modo de escuta quando não está transmitindo dados através da rede. O nó, após encerrar uma transmissão, passa a dormir e só acorda em períodos definidos para executar as demais tarefas da rede (VASSEUR *et al.*, 2010).

Entre as camadas de enlace e de rede, existe uma camada de adaptação. Essa camada foi criada para possibilitar a utilização do protocolo IPv6 nos dispositivos da IoT. Com a utilização do protocolo IPv6 é possível utilizar toda a infraestrutura que já existe na internet. Com a adoção da camada de adaptação não é necessário modificar o protocolo IPv6 para que ele seja utilizado pela IoT. Alguns dos protocolos presentes nessa camada são: o IPv6 over Low Power Wireless Personal Area Networks (6LoWPAN), IPv6 sobre Bluetooth Low Power e o 6Lo. O Quadro 1.4 apresenta uma síntese sobre os protocolos de comunicação mais utilizados na IoT.

Quadro 1.4 | Síntese dos principais protocolos de comunicação utilizados na IoT

	6LoWPAN	ZigBee	Bluetooth Low Energy	Bluetooth Clássico	RFID	NFC	WiFi	Wi-Fi Low Energy	LoRa
Padrão	IEEE 802.15.4	IEEE 802.15.4	IEEE 802.15.1	IEEE 802.15.1	RFID	ISO/IEC 14443	IEEE 802.11	IEEE 802.11	LoRa Alliance
Frequência de Operação	subGHz e 2,4GHz	2,4GHz	2,4GHz	2,4GHz	subGHz	subGHz	subGHz, 2,4GHz e 5GHz	subGHz	subGHz
Tipo de Rede	WPAN	WPAN	WPAN	WPAN	Proximidade	Rede P2P	WLAN	WPAN	LPWAN
Topologia	Estrela e Mista	Estrela, Mista e Cluster	Estrela e broadcasting	Mestre/Escravo Piconet	P2P	P2P	Estrela e P2P	Estrela e árvore	Estrela
Consumo de Energia	baixo	baixo	baixo	médio	muito baixo	muito baixo	alto	baixo	baixo
Alcance	10-100m	10-100m	15-30m	10-100m	200m	1cm-1m	10-100m	10-100m	até 15km
Custo	\$	\$	\$\$	\$\$\$	\$	\$	\$\$\$\$	\$\$	\$\$
Aplicações	Monitamento e controle via internet	Monitamento residencial e industrial	Comunicação entre dispositivos pessoas/ Beacon	Comunicação entre dispositivos pessoas	Rastreamento/ Inventário	Pagamento/ Acesso	Rede pessoal/ computadores	Dispositivos pessoas	Monitoramento de grandes áreas

Fonte: adaptado de Al-sarawi *et al.* (2017, [s.p.]).

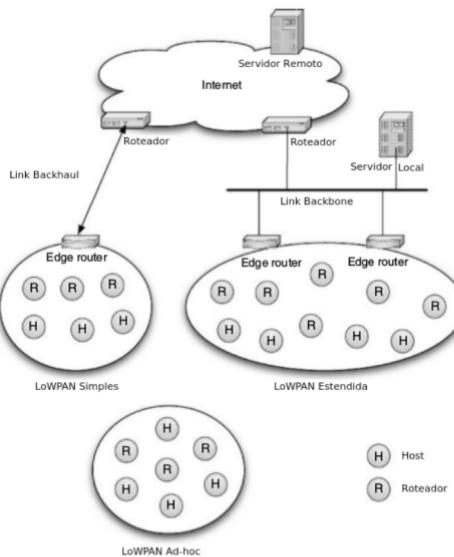
O 6LoWPAN é o protocolo mais utilizado e mais importante presente na camada de adaptação (SETHI; SARANGI, 2017). Com o emprego do 6LoWPAN é possível utilizar o protocolo IPv6 para a comunicação dos dispositivos da IoT. Essa é uma grande vantagem do 6LoWPAN, pois é possível aproveitar a estrutura já existente da internet e o grande espaço de endereçamento disponível no protocolo IPv6 (SHELBY; BORMANN, 2011).

Para que o IPv6 possa ser utilizado de forma eficiente sobre as plataformas de baixo consumo energético em redes sem fio é necessário que sejam realizadas adaptações de algumas camadas e otimização de protocolos (SHELBY; BORMANN, 2011). O papel dessa camada é possibilitar o encapsulamento dos pacotes IPv6 que são recebidos pela camada de rede e realizar a compressão do cabeçalho IPv6 e a fragmentação dos pacotes. Essas ações têm como objetivos reduzir o consumo de energia e possibilitar a integração das camadas inferiores (definidas pelo IEEE 802.15.4) e as camadas superiores do protocolo IPv6.

Em uma rede construída utilizando-se o protocolo 6LoWPAN podemos conectar os dispositivos de três modos diferentes, sendo os seguintes: Ad-Hoc, simples e estendida. A Figura 1.10 ilustra essas três configurações. As LoWPAN simples são redes interconectadas via internet por meio de um roteador de borda (*Border Router*). Na configuração LoWPAN estendida, a rede local é conectada à internet por meio de múltiplos roteadores de borda,

podendo ser ligado a um backbone. Diferentemente da LoWPAN simples e estendida, as redes LoWPAN Ad hoc, não são conectadas à internet, assim as mensagens são trocadas apenas dentro de uma rede local.

Figura 1.10 | Diferentes configurações disponíveis para o protocolo 6LoWPAN



Fonte: adaptada de Shelby e Bormann (2011, [s.p.]).

O Bluetooth *Low Energy* (BLE), também conhecido como Bluetooth *Smart*, é uma variação do protocolo Bluetooth tradicional. O BLE foi desenvolvido para aplicações que requerem a transmissão de pequenas quantidades de dados e com baixas taxas de transmissão, realizadas por um curto período de tempo e que devem priorizar o consumo reduzido de energia (NIEMINEN *et al.*, 2015). O IPv6 sobre Bluetooth *Low Power* utiliza, praticamente, as mesmas técnicas de compressão do cabeçalho IPv6 realizadas pelo 6LoWPAN. Entretanto, devido às características de construção do BLE, o processo de fragmentação dos pacotes não é utilizado pelo IPv6 sobre Bluetooth *Low Power*.

Os protocolos mais conhecidos para o roteamento das mensagens em uma rede IoT são: o *Routing Protocol for Low-Power and Lossy Networks* (RPL), o *Cognitive RPL* e o *Channel-Aware Routing Protocol* (CARP) (*Networking Protocols and Standards for Internet of Things*).

O RPL é o protocolo mais utilizado para realizar o roteamento de pacotes. Ele foi desenvolvido para ser utilizado pelas *Low-Power and Lossy Network*

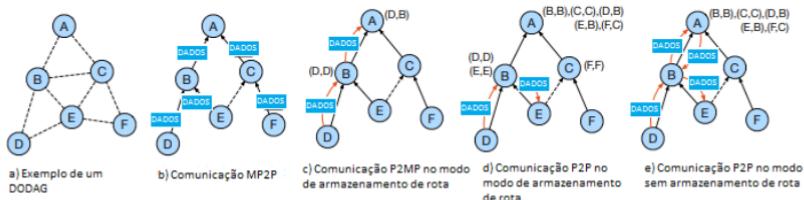
(LLN), assim, apresenta características distintas dos protocolos de roteamento tradicionais.

Assimile

As LLN são redes compostas por elementos com disponibilidade reduzida de recursos e que operam com baixas taxas de transferências de mensagens, grande quantidade de perdas de pacotes e baixo consumo de energia (BORMANN; ERSUE; KERANEN, 2014).

O RPL é baseado em vetores de distância e o roteamento é realizado por meio de grafos acíclicos orientados a destino (DODAG do inglês, *Destination Oriented Directed Acyclic Graphs*). Um grafo acíclico corresponde a um grafo que não possui ciclos entre os nós. Um grafo (DODAG) é um grafo acíclico em que todos os nós apontam (são direcionados) para um nó raiz (*border router*). A Figura 1.11a apresenta um exemplo de um grafo DODAG. É possível ver que esse grafo, que representa a rede, possui um nó raiz e os demais nós são direcionados (apresentam setas) para esse nó raiz. Diferentemente do roteamento convencional que mantém tabelas com as rotas já estabelecidas, o protocolo RPL pode encontrar as rotas para a entrega de pacotes no momento em que é solicitado o envio da mensagem. É o tipo de configuração do RPL que indica como as rotas devem ser construídas. No primeiro modo de configuração, o nó raiz é o único nó a conhecer toda a estrutura da DODAG (Figura 1.11c) e os demais nós são responsáveis apenas por manter informações sobre os nós pais. Esse tipo de formação da RPL é o mais comum. No segundo modo de operação (Figura 1.11d), os demais nós guardam informações sobre os nós pais e filhos, assim, pode ocorrer a comunicação direta entre esses nós, sem a necessidade de enviar a mensagem até o nó raiz. Essas diferentes configurações são fundamentais para redes LLN, pois, a fim de economizar energia, quando os nós não estão transmitindo eles passam a “dormir”, o que dificulta manter as rotas já estabelecidas. Além disso, os nós de uma LLN estão mais sujeitos a falhas, o que obrigaria a atualizações constantes das tabelas de roteamento (IOVA *et al.*, 2016).

Figura 1.11 | Exemplo de construção do roteamento RPL.



Fonte: adaptada de (IOVA *et al.* 2016).

Por padrão, cada nó presente na rede mantém múltiplos nós pais em relação ao nó raiz. Por exemplo, na Figura 1.11a os nós pais do nó E poderiam ser os nós B e C. Entretanto, o nó E escolhe uma rota preferencial para envio dos pacotes, por exemplo, pelo nó B e deixa o pai C como rota de backup para o nó raiz. Como o nó raiz recebe dados de vários nós, esse esquema é conhecido como comunicação *multipoint-to-point*, sendo um tipo de *conexão many-to-one* (BORMANN; ERSUE; KERANEN, 2014). No RPL também existe a possibilidade de comunicação do nó raiz com os demais nós presentes na DODAG. Esse tipo de comunicação em que temos um nó transmitindo para vários é conhecido como comunicação *point-to-multipoint* (P2MP), também sendo chamado de *one-to-many*. Na comunicação P2MP temos múltiplos caminhos partindo de uma única localização para múltiplos destinos. Um exemplo é a comunicação do nó A (*border router*) com os nós B e C. Outro tipo de comunicação que pode existir no protocolo RPL é do tipo *point-to-point* (P2P), conhecida como uma conexão *one-to-one*. Esse tipo de comunicação ocorre quando os nós podem realizar a troca de mensagens um diretamente com o outro. Todos esses tipos de comunicação realizados pelos protocolos RPL são coordenados por meio do envio de mensagens que indicam qual o tipo de comunicação deve ser realizado (BORMANN; ERSUE; KERANEN, 2014). O Quadro 1.5 apresenta uma síntese para os modos de construção de roteamento utilizando o RPL.

Quadro 1.5 | Modos de roteamento utilizando a RPL

Comunicação	Tipo de Mensagens Trocadas	Sentido do Fluxo de Mensagens
Multipoint-to-Point (MP2P)	Nós enviam mensagens para o nó Raiz	Baixo para cima
Point-to-Multipoint (P2MP)	Nó raiz envia mensagem para outros nós	Cima para baixo
Point-to-Point (P2P)	Troca de mensagens entre nós (sem nó raiz)	Baixo para cima e cima para baixo

Fonte: adaptado de Sobral *et al.* (2016, [s.p.]).

Outro protocolo utilizado nessa camada é o *Cognitive RPL* (CORPL). O CORPL também utiliza o grafo DODAG para construir a rede e possibilitar o roteamento dos pacotes. Entretanto, em vez dos nós manterem apenas informações sobre os seus pais, cada nó mantém informações sobre as rotas até o nó raiz. Essas rotas são modificadas assim que é identificada alguma mudança nos nós vizinhos. Esse protocolo foi desenvolvido para utilização

em redes cognitivas (AIJAZ; AGHVAMI, 2015), as quais correspondem as redes dotadas da capacidade de perceber alterações e adaptar a essas mudanças (SOUZA *et al.*, 2010).

O CARPL é um protocolo desenvolvido para ser utilizado em aplicações subaquáticas. Como a comunicação é realizada utilizando algumas das características das redes LLN, o CARPL pode ser utilizado na camada de rede da IoT. A escolha sobre qual rota deve ser utilizada para comunicação é realizada por meio da avaliação do histórico da qualidade da conexão existente entre os nós (BASAGNI *et al.*, 2015).

Os protocolos da camada de transporte são aqueles que garantem a comunicação entre dois dispositivos. Existem dois principais protocolos: *Transmission Control Protocol* (TCP) e o *User Datagram Protocol* (UDP).

O TCP é orientado a conexões, ou seja, é estabelecida uma conexão entre os dispositivos antes que os dados da mensagem comecem a ser enviados. Além disso, a cada pacote recebido por um dos dispositivos é enviada uma mensagem de recebimento, que corresponde que o envio do pacote de dados foi feito corretamente. Caso seja identificada alguma falha no recebimento dos pacotes, eles são retransmitidos. Assim, o TCP garante que todos os pacotes enviados foram corretamente recebidos pelo destinatário. Já no UDP, não existe a garantia do correto recebimento dos dados, pois não existe o estabelecimento prévio de uma conexão e não há a confirmação de recebimento correto da mensagem (TANENBAUM, 2007). A utilização do protocolo UDP ocorre em redes que necessitam de maior velocidade e eficiência na troca dos dados, já em aplicações em que a confiabilidade é primordial, é preferível utilizar o protocolo TCP.

Os principais representantes da camada de aplicação são os protocolos: *Constrained Application Protocol* (CoAP) e o *Message Queue Telemetry Transport* (MQTT).

O CoAP é similar ao *HyperText Transfer Protocol* (HTTP), entretanto, existem diferenças que o tornam útil para a utilização em redes IoT. Diferentemente do HTTP, o CoAP utiliza UDP para a comunicação entre os dispositivos. Mesmo utilizando o UDP, o CoAP implementa técnicas para assegurar o envio e recebimento correto das mensagens. Como no HTTP, o CoAP utiliza as funções *GET*, *PUT*, *PUSH* e *DELETE* para realizar requisições. O CoAP realiza a compressão dos cabeçalhos, ele possui controle de congestionamento e suporta comunicação *multicast*. Outra característica do CoAP é a utilização do *Efficient XML Interchanges* (EXI) para formatar os dados. O EXI é uma versão mais eficiente do HTML/XML que possibilita a utilização em dispositivos com baixa capacidade de armazenamento e

memória. Para existir a comunicação entre os protocolos HTTP e CoAP é necessário que exista um proxy HTTP-CoAP. É o proxy que possibilita que uma rede LLN utilizando CoAP possa se comunicar com a internet. Assim, mesmo que inspirado no HTTP, o CoAP foi criado para ser utilizado em dispositivos de recursos limitados (SHELBY *et al.*, 2014).

O MQTT é um protocolo de troca de mensagens entre máquinas (M2M), ele é composto por 3 componentes principais: os *publishers* (publicadores), os *subscribers* (assinadores) e um *broker*. Os *publishers* são os dispositivos responsáveis por publicar (enviar) as mensagens para o *broker*. Um exemplo de *publisher* pode ser um sensor de temperatura. Esse sensor publica o valor de temperatura lido no *broker*. O *broker* é quem vai receber as mensagens dos *publisher* e vai disponibilizá-las aos *subscribers*. Os *subscribers* são os dispositivos que vão consumir as informações disponibilizadas pelos *publishers* e que estão disponíveis no *broker*. Os *subscribers* podem ser, por exemplo, um smartphone que deseja ler a temperatura de um sensor que esteja na cozinha. Para obter essa informação, o *subscriber* assina um tópico correspondente ao sensor de temperatura que esteja na cozinha. O tópico especifica, por exemplo, a localização e de qual sensor ele deseja obter a medida. Quando uma modificação nesse dado (tópico) ocorre o *broker* informa ao *subscriber* que está inscrito nesse tópico. Assim, os *publishers* que são, normalmente, os dispositivos com recursos limitados podem gerar os dados necessários, enviá-los ao *broker* e voltarem a dormir e, portanto, economizar energia (KARAGIANNIS *et al.*, 2015).

Além das definições de protocolos, é necessário que sejam definidas as interfaces de interação com usuário. A interface pode ser definida como a parte do sistema em que o usuário mantém algum tipo de contato, seja por meio de software ou hardware, podendo ser de forma ativa ou passiva (PRATES; BARBOSA, 2003). Sendo assim, essa é uma das partes mais importantes para a construção de um sistema. Isso ocorre porque todos os códigos e protocolos que usamos para construir uma aplicação estão invisíveis ao usuário, assim, é por meio da interface que ele terá contato com todas as funcionalidades do sistema criado.

A interface com o usuário não precisa, necessariamente, ocorrer por meio de telas ou softwares. Ela pode ocorrer por meio de botões, luzes ou gestos, por exemplo. A escolha adequada do modo como o usuário utilizará o sistema é essencial para garantir uma experiência agradável e confortável de utilização. Essa escolha também deve ser realizada com base na aplicação. Por exemplo, em um sistema de alarme de incêndio, podemos utilizar *Short Message Service* (SMS) para alertar os administradores do sistema sobre o ocorrido, para enviar um e-mail ou outras notificações via aplicativos no

smartphone. Em aplicações de monitoramento, como em rastreamento de veículos, podemos criar aplicativos para smartphones, tablets ou computadores pessoais que possibilitem o acompanhamento do trajeto dos veículos monitorados. Já em aplicações que necessitem de acionamento remoto, por exemplo, apagar e acender luzes ou abrir e fechar portas, devemos também deixar a possibilidade de utilização de interfaces físicas. Essas interfaces físicas podem ser interruptores ou chaves de contato, por exemplo. Além disso, em alguns momentos em que a conexão com internet não está disponível é necessário que essas funcionalidades estejam acessíveis para o usuário. A integração entre todas essas interfaces é a chave para construir aplicações.

Exemplificando

Um exemplo de utilização bem-sucedida de integração de diferentes interfaces pode ser encontrado no Google Nest (HERNANDEZ et al., 2014). Nele existe a integração de interfaces físicas, de software e telas de aplicações em smartphones e computadores. O Google Nest foi, inicialmente, desenvolvido para ser um termostato inteligente, capaz de aprender com as preferências do usuário sobre qual a temperatura ideal em cada situação e, por meio desse aprendizado, ajustar automaticamente a temperatura da residência. Após a criação desse termostato outros produtos foram inseridos para a realização da automação residencial. Hoje, o Google Nest conta com produtos para a detecção automática e alerta de incêndio, câmeras IP e sensores de presença para a segurança e uma central inteligente de monitoramento de todo o sistema.

Manter os sistemas invisíveis, ou seja, sem que comprometam ou atrapalhem as nossas tarefas diárias é uma das mais importantes funções das interfaces (WEISER; BROWN, 1997). Portanto, assegurar que a escolha da interface esteja alinhada com a melhor experiência do usuário e com a melhor usabilidade e o melhor aproveitamento da aplicação é essencial para o desenvolvimento da IoT.

Chegamos ao final desta unidade. Percorremos um longo caminho em busca de conhecer melhor a Internet das Coisas. Ao final dessa caminhada, temos conhecimento sobre os principais aspectos construtivos da IoT, como definições, arquitetura, protocolos, sensores e atuadores, segurança e várias outras questões que são fundamentais para o desenvolvimento de aplicações robustas e eficientes. Vamos continuar aprendendo, pois terminamos apenas a primeira etapa de nossa jornada.

Sem medo de errar

Após você indicar os sensores e atuadores, mostrar quais as questões de segurança mais importantes a serem observadas, chegou a hora de completar o seu relatório de levantamento de requisitos para a automação da casa modelo. Os diretores esperam que você complete o relatório adicionando sugestões de interfaces e protocolos a serem utilizados para a construção da casa modelo. Além de indicar quais os protocolos a serem empregados em cada uma das camadas, eles esperam que você justifique a suas escolhas.

A casa modelo em que você está trabalhando possui quartos, sala, banheiro, cozinha, garagem e escritório. As interfaces e protocolos escolhidos devem possibilitar a integração de toda a residência e facilitar a vida dos moradores. Além disso, devem ser observadas as características de cada um deles e, assim, alinhar a escolha com as necessidades da aplicação.

Para escolher corretamente os protocolos em cada uma das camadas, você deve observar as características de cada uma das opções. Assim, comparando essas características de construção com as necessidades da aplicação é possível indicar qual é a combinação desses protocolos que melhor atende às necessidades dos clientes.

O principal ponto a ser observado para a escolha das interfaces é indicar aquelas que facilitam a vida dos moradores e não gerem transtornos ou dificuldades de utilização. Logo, a integração eficiente entre diferentes interfaces é essencial para o sucesso da aplicação.

O Quadro 1.6 apresenta um exemplo de utilização de protocolos em cada uma das camadas. Lembrando que essa é apenas uma sugestão e resposta. Podem existir outras combinações de protocolos e interfaces que garantem os resultados esperados.

Quadro 1.6 | Exemplo de utilização de protocolos em cada uma das camadas

Camada	Protocolo	Justificativa
Física/Enlace	IEEE 802.15.4	Garante que os requisitos de consumo de energia e eficiente utilização de memória e processamento sejam alcançados. Como queremos manter um baixo custo dos equipamentos, eles, normalmente, devem possuir recursos limitados.
Adaptação	6LoWPAN	Pois o 6LoWPAN possibilita a utilização do protocolo IPv6 em dispositivos com recursos limitados. Assim, é possível utilizar toda a infraestrutura da internet para comunicação global em nossa aplicação.

Camada	Protocolo	Justificativa
Rede	IPv6+RPL	Pois o RPL possibilita o roteamento das mensagens com um reduzido consumo de energia. Como nem todos os nossos dispositivos estarão conectados à rede elétrica, essa é uma característica importante.
Transporte	UDP	Como queremos eficiência e velocidade no transporte das mensagens, o UDP é o protocolo mais indicado.
Aplicação	CoAP	Como queremos integrar as diferentes interfaces (smartphones/pc/física), possibilitar fácil tradução para HTTP e RESTfull é uma característica interessante para nossa aplicação.

Fonte: elaborado pelo autor.

O principal ponto a ser observado para a escolha das interfaces é indicar aquelas que facilitam a vida dos moradores e não gerem transtornos ou dificuldades de utilização. Logo, a integração eficiente entre diferentes interfaces é essencial para o sucesso da aplicação.

O Quadro 1.6 apresenta um exemplo de utilização de protocolos em cada uma das camadas. Lembrando que essa é apenas uma sugestão e resposta. Podem existir outras combinações de protocolos e interfaces que garantem os resultados esperados.

Para a escolha das interfaces é necessária a união de elementos físicos como interruptores, chaves de abertura de portas e controles remotos com interfaces de software. É necessário que exista a integração dos softwares em diferentes plataformas. Por exemplo, deve ser possível acessar o sistema por meio de smartphones, tablets e computadores. As notificações de eventos como detecção de incêndio e invasores devem ser enviadas tanto por SMS quanto via aplicativos dos celulares. As interfaces físicas devem estar integradas aos softwares a fim de não interferirem no funcionamento normal da residência.

Assim, com a escolha dos protocolos e interfaces, você acaba de completar o relatório de levantamento de requisitos para o seu primeiro trabalho de domótica.

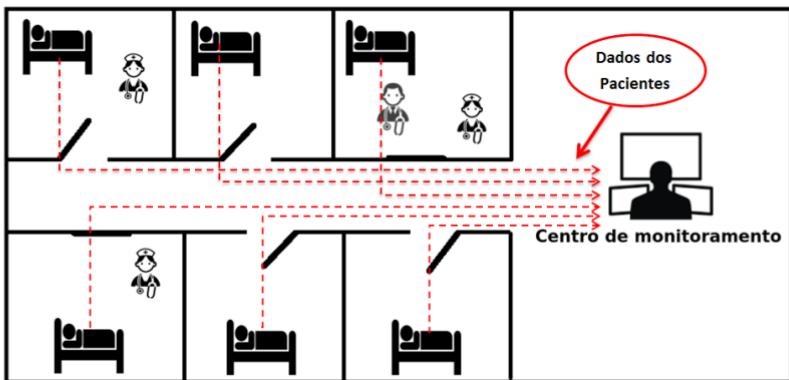
Com certeza, os diretores gostaram bastante de seu trabalho. Agora é só passar no setor de recursos humanos e assinar o seu contrato. Parabéns!

Avançando na prática

Automação de uma enfermaria

Nessa aplicação, você foi contratado para realizar a automação de uma enfermaria. A enfermaria é o local de monitoramento de pacientes. Para o acompanhamento desses pacientes é necessário monitorar os sinais vitais e as ações dos pacientes que estão alojados nessa enfermaria (AL-FUQAH et al., 2015). Para isso, são instalados sensores nos pacientes e nos quartos em que estão hospedados. Esses sensores enviam os sinais para uma central para que possam ser analisados e possíveis comportamentos anormais dos pacientes consigam ser identificados e, prontamente, tratados.

Figura 1.12 | Automação de uma enfermaria – Monitoramento de pacientes



Fonte: elaborada pelo autor.

Resolução da situação-problema

O Quadro 1.7 apresenta um cenário dessa aplicação.

Quadro 1.7 | Aplicação de automação em enfermaria

Camada	Protocolo	Justificativa
Física/Enlace	IEEE 802.15.4	Pois os dispositivos de monitoramento acoplados ao corpo dos pacientes devem ser alimentados por baterias. Assim, devem garantir um consumo eficiente de energia.

Camada	Protocolo	Justificativa
Adaptação	6LoWPAN	Para que os dados possam ser compartilhados entre diferentes médicos de maneira eficiente via internet.
Rede	IPv6+RPL	A fim de possibilitar a integração com a internet e roteamento eficiente da informação.
Transporte	TCP	Os dados devem ser corretos, não podemos receber mensagens que contêm informações diferentes das medidas pelos sensores.
Aplicação	MQTT	Como possibilita a integração com o TCP e o cenário entre publish (sensores dos pacientes/quartos) e subscribes (central) está bem definido, essa é uma alternativa interessante.

Fonte: elaborado pelo autor.

Para a interface é interessante a união entre as partes físicas dos sensores/atuadores e os softwares existentes. É necessário que ambas funcionem corretamente e que caso ocorra uma falha em uma das interfaces isso não comprometa o funcionamento da outra, pois garantir a confiabilidade do sistema é primordial nessa aplicação.

Faça valer a pena

1. Quando estamos criando uma aplicação em IoT, uma das principais características a ser observada corresponde ao desenvolvimento de uma interface amigável e eficiente. Para garantir a eficiência dessa interface é necessário que sua utilização seja natural para o usuário. Segundo Mark Weiser e Jhon Brown, os dispositivos devem estar em todos os lugares e por isso é melhor que eles não nos atrapalhem (WEISER; BROWN, 1997). Essa afirmação de Weiser e Brown é muito conhecida, pois traduz um dos principais conceitos relativos ao desenvolvimento de aplicações IoT.

Assinale a alternativa que melhor traduz a fala de Weiser e Brown no contexto das interfaces para a IoT.

- a. As interfaces devem possibilitar o agrupamento do maior número de sensores possíveis, pois assim eles podem ser colocados em vários lugares.
- b. Todas as interfaces devem priorizar a conexão de inúmeros dispositivos.
- c. Quando utilizamos interfaces diferentes, precisamos interligar todos os sensores.

- d. As interfaces devem ser construídas de modo a proporcionar ao usuário uma interação natural com a aplicação.
- e. Interfaces devem ser utilizadas a fim de integrar a maior quantidade de protocolos, pois a união de diferentes dispositivos é uma das características da IoT.

2. Existem várias diferentes definições sobre a divisão em camadas para a IoT. Vários autores indicam uma divisão diferente sobre os nomes e protocolos utilizados. Yang *et al.* (2011) sugere a divisão em 3 camadas, Khan *et al.* (2012), White, Fish e Pooch (2017) apresentam a arquitetura da IoT em 5 camadas, enquanto Hanes *et al.* (2017) a divisão em 7. Como pôde ser visto, assim como ocorre com as redes de computadores, existem vários modelos utilizados para sintetizar os relacionamentos entre os protocolos. Apesar dessas diferentes hierarquias, a utilização de uma subcamada de adaptação é apresentada em todos os trabalhos.

A principal utilidade da subcamada de adaptação para o sucesso da IoT é:

- a. que por meio dessa camada é possível utilizar o protocolo IPv6 em dispositivos com disponibilidade limitada de recursos.
- b. possibilitar a utilização do MQTT como protocolo de troca de mensagens.
- c. garantir a integração entre as camadas de rede e aplicação.
- d. permitir adaptação dos computadores pessoais para utilizarem os protocolos presentes na IoT.
- e. integrar diferentes dispositivos de diferentes fabricantes.

3. O Internet Engineering Task Force (IETF) iniciou, em 2005, o desenvolvimento de um protocolo que possibilitaria a utilização do IPv6 em dispositivos de baixo consumo de energia, que possuem baixas taxas de transmissão de dados e que realizam tarefas simples (SHELBY; BORMANN, 2011). Em 2007, a primeira versão desse protocolo foi lançada. Atualmente, esse protocolo, conhecido como 6LoWPAN, é o mais utilizado na subcamada de adaptação. A grande adoção desse protocolo se deve ao fato de possibilitar a integração eficiente dos dispositivos da IoT com a internet, por meio da utilização do IPv6.

Sobre o 6LoWPAN, análise as afirmativas a seguir.

- I. O 6LoWPAN implementa as camadas físicas e de enlace apresentadas pelo IEEE 802.15.4.
- II. A fragmentação dos pacotes e redução do cabeçalho IPv6 são tarefas realizadas pelo 6LoWPAN.

- III. O 6LoWPAN é o único protocolo que permite a integração dos dispositivos IoT com a internet.
- IV. É necessário modificar o protocolo IPv6 para que ele seja utilizado em conjunto com o 6LoWPAN.

Considerando as características do protocolo 6LoWPAN, assinale a alternativa correta.

- a. Apenas I e IV estão corretas.
- b. Apenas I e III estão corretas.
- c. Apenas I e II estão corretas.
- d. Apenas I, II e IV estão corretas.
- e. As afirmativas I, II, III e IV estão corretas.

Referências

- ADAMS, J. T. An introduction to IEEE STD 802.15. 4. In: 2006 IEEE AEROSPACE CONFERENCE, 2006, [s.l.]. **Anais** [...]. [S.I.]: IEEE, 2006. 8 p.
- AIJAZ, A.; AGHVAMI, A. H. Cognitive machine-to-machine communications for Internet-of-Things: A protocol stack perspective. **IEEE internet of things journal**, v. 2, n. 2, p. 103-112, 2015.
- AL-FUQABA, A. et al. Internet of things: a survey on enabling technologies, protocols, and applications. **Communications Surveys & Tutorials**, [s.l.], v. 17, n. 4, p. 2.347-2.376, 2015.
- AL-SARAWI, S. et al. Internet of Things (IoT) communication protocols. In: 2017 8th INTERNATIONAL CONFERENCE ON INFORMATION TECHNOLOGY, 8., 2017, [s.l.]. **Anais** [...]. [S.I.]: IEEE, 2017. p. 685-690.
- ASHTON, K. et al. That ‘internet of things’ thing. **RFID journal**, v. 22, n. 7, p. 97-114, 2009.
- BALBINOT, A.; BRUSAMARELLO, V. **Instrumentação e fundamentos de medidas–volumes I e II**. [S.I.]: Editora LTC, 2007.
- BASAGNI, S. et al. CARP: A channel-aware routing protocol for underwater acoustic wireless networks. **Ad Hoc Networks**, v. 34, p. 92-104, 2015.
- BHUYAN, M. **Intelligent instrumentation**: principles and applications. [S.I.]: CRC Press, 2010.
- BORMANN, C.; ERSUE, M.; KERANEN, A. **RFC 7228**: Terminology for Constrained-Node Networks. IETF Request For Comments, 2014.
- CIRINO, L. **Atuadores de máquinas e equipamentos**. 2018. Disponível em: <https://www.manutencaoomfoco.com.br/atuadores-de-maquinas-e-equipamentos/>. Acesso em: 11 jul. 2019.
- COHEN, S.; BATAILLE, L. R.; MARTIG, A. K. Enabling breakthroughs in Parkinson’s disease with wearable technologies and big data analytics. **Mhealth**, v. 2, 2016.
- CONNECTED WORLD. **Urbanization and transportation**. 2019. Disponível em: <https://connectedworld.com/may-urbanization-and-transportation/>. Acesso em: 11 jul. 2019.
- DAMERI, R. P. et al. **Smart city implementation**. Cham: Springer International Publishing AG, 2017.
- DE, S. et al. An internet of things platform for real-world and digital objects. **Scalable Computing: Practice and Experience**, v. 13, n. 1, p. 45-58, 2012.
- DOUKAS, C. **Building Internet of Things with the Arduino**. [S.I.]: CreateSpace Independent Publishing Platform, 2012.
- EVANS, D. The internet of things: how the next evolution of the internet is changing everything. **CISCO white paper**, v. 1, p. 1-11, 2011.
- FARAHANI, S. **ZigBee wireless networks and transceivers**. [S.I.]: Newnes, 2011.

FORTINO, G.; TRUNFIO, P. (Ed.). **Internet of things based on smart objects: technology, middleware and applications.** [S.l.]: Springer Science & Business Media, 2014.

GAUR, A. et al. Smart city architecture and its applications based on IoT. **Procedia computer science**, v. 52, p. 1089-1094, 2015.

GAY, W. **Advanced Raspberry Pi**: Raspbian Linux and GPIO Integration. Apress: [s.l.], 2018.

GILCHRIST, A. **Industry 4.0: the industrial internet of things.** [S.l.]: Apress, 2016.

GUTIERREZ, J. A.; CALLAWAY, E. H.; BARRETT, R. L. **Low-rate wireless personal area networks: enabling wireless sensors with IEEE 802.15. 4**. IEEE Standards Association, 2004.

HANES, D. et al. **IoT fundamentals**: networking technologies, protocols, and use cases for the internet of things. [S.l.]: Cisco Press, 2017.

HARIYAWAN, M. Y.; GUNAWAN, A.; PUTRA, E. H. Wireless sensor network for forest fire detection. **Telkomnika**, v. 11, n. 3, p. 563, 2013.

HASSAN, W. H. et al. Current research on Internet of Things (IoT) security: a survey. **Computer Networks**, [s.l.], v. 148, p. 283-294, 2019.

HERNANDEZ, G. et al. Smart nest thermostat: A smart spy in your home. In: BLACK HAT USA, 2014, Las Vegas – NV. **Materials** [...], Las Vegas – NV: Black hat, 2014. p. 1-8.

HERSENT, O.; BOSWARTHICK, D.; ELLOUMI, O. **The internet of things: key applications and protocols.** [S.l.]: John Wiley & Sons, 2011.

IBM. **Internet of Things blog**. [s.d.]. Disponível em: <https://www.ibm.com/blogs/internet-of-things/>. Acesso em: 11 jul. 2019.

INSTRUMENTS, T. **LM35 Precision Centigrade Temperature Sensors.** [S.l.]: LM35 datasheet, 1999.

IOVA, O. et al. Rpl: The routing standard for the internet of things... or is it? **IEEE Communications Magazine**, v. 54, n. 12, p. 16-22, 2016.

JUNIOR, J. J. A. M.; JUNIOR, S. L. S. **LDR e sensores de luz ambiente: funcionamento e aplicações.** [s.l.], 2013.

JURVETSON, S. **Google's Lexus RX 450h Self-Driving Car.** 2012. Disponível em: https://commons.wikimedia.org/wiki/File:Google%27s_Lexus_RX_450h_Self-Driving_Car.jpg. Acesso em: 11 jul. 2019.

KARAGIANNIS, V. et al. A survey on application layer protocols for the internet of things. **Transaction on IoT and Cloud computing**, v. 3, n. 1, p. 11-17, 2015.

KAUR, N.; MONGA, S. Comparisons of wired and wireless networks: a review. **International Journal of Advanced Engineering Technology**, [s.l.], v. 5, n. 2, p. 34-35, 2014.

KHAN, R. et al. Future internet: the internet of things architecture, possible applications and key challenges. In: INTERNATIONAL CONFERENCE ON FRONTIERS OF INFORMATION TECHNOLOGY, 10., 2012, [s.l.]. **Anais** [...]. [S.l.]: IEEE [...], 2012. p. 257-260.

- KIM, T.; RAMOS, C.; MOHAMMED, S. **Smart city and IoT**. [s.l.], 2017.
- KURUNATHAN, H. *et al.* IEEE 802.15. 4e in a nutshell: Survey and performance evaluation. *IEEE Communications Surveys & Tutorials*, v. 20, n. 3, p. 1989-2010, 2018.
- KYUNG, C. *et al.* (Ed.). **Smart sensors and systems**: innovations for medical, environmental, and IoT applications. [S.l.]: Springer, 2016.
- LEWIS, D.; LEIBRAND, S.; #OPENAPS COMMUNITY. Real-world use of open source artificial pancreas systems. *Journal of diabetes science and technology*, v. 10, n. 6, p. 1411-1411, 2016.
- LOUREIRO, A. A. F. *et al.* Redes de sensores sem fio. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES, 21., 2003, Natal. *Anais* [...]. Natal: SBRC, 2003. p. 179-226. Disponível em: <https://homepages.dcc.ufmg.br/~linnyer/minicursoSBRC2003.pdf>. Acesso em: 12 jul. 2019.
- MAHMOOD, Z. (Ed.). **Guide to ambient intelligence in the IoT Environment**: principles, technologies and applications. [S.l.]: Springer, 2019.
- NIEMINEN, J. *et al.* **RFC 7668-IPv6 over Bluetooth Low Energy**. Fremont, CA: IETF, 2015.
- OLSSON, J. 6LoWPAN demystified. *Texas Instruments*, [s.l.], v. 13, 2014.
- PARK, M. IEEE 802.11 ah: sub-1-GHz license-exempt operation for the internet of things. *IEEE Communications Magazine*, v. 53, n. 9, p. 145-151, 2015.
- PETRUZELLA, F. **Motores elétricos e açãoamentos**: série Tekne. [S.l.]: Bookman Editora, 2013.
- PRATES, R. O.; BARBOSA, S. D. J. Avaliação de interfaces de usuário—conceitos e métodos. In: CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 23., 2003, Campinas-SP. *Anais* [...]. Campinas: SBC, 2003. p. 28.
- PRESSER, M. **Inspirando a Internet das Coisas**. Edição Brasileira do Comic Book, 2011. Disponível em: https://iotcomicbook.files.wordpress.com/2013/10/iot_comic_book_special_br.pdf. Acesso em: 11 jul. 2019.
- ROSER, C. **Industry 4.0**. 2015. Disponível em: https://commons.wikimedia.org/wiki/File:Industry_4.0.png. Acesso em: 11 jul. 2019.
- SANCHEZ, L. *et al.* SmartSantander: IoT experimentation over a smart city testbed. *Computer Networks*, [s.l.], v. 61, p. 217-238, 2014.
- SANTOS, B. P. *et al.* Internet das coisas: da teoria à prática. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES E SISTEMAS DISTRIBUÍDOS, 2016, [s.l.]. *Minicursos* [...]. [S.l.]: SBRC, 2016.
- SETHI, P.; SARANGI, S. R. Internet of things: architectures, protocols, and applications. *Journal of Electrical and Computer Engineering*, [s.l.], v. 2017, 2017.
- SHELBY, Z. *et al.* **RFC 7252**: The constrained application protocol (CoAP). Internet Engineering Task Force, 2014.

SHELBY, Z.; BORMANN, C. **6LoWPAN**: the wireless embedded Internet. Hoboken: John Wiley & Sons, 2011.

SOBRAL, J. V. V. *et al.* Routing Protocols for Low Power and Lossy Networks in Internet of Things Applications. **Sensors**, v. 19, n. 9, p. 2144, 2019.

SOLON, O. Team of hackers take remote control of Tesla Model S from 12 miles away. **The Guardian**, Londres, 2016.

SORNIN, N. *et al.* **Lorawan specification**. LoRa alliance, 2015.

SOUSA, M. P. *et al.* Redes cognitivas: um novo paradigma para as comunicações sem fio. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES E SISTEMAS DISTRIBUÍDOS, 28., 2010, Gramado-RS. **Minicursos** [...]. Gramado-RS: Sociedade Brasileira de Computação, 2010. p. 153-197.

SUN, A. C.; CHEN, C.; LEE, C. In-circuit programming architecture with ROM and flash memory. **U.S. Patent**, n. 5, 901,330, 4 maio 1999.

TANENBAUM, A. S. **Redes de computadores**. 4. ed. Amsterdam: Vrije Universiteit, 2003.

TANENBAUM, A. S. **Redes de computadores**: PLT. Rio de Janeiro: Campus, 2007.

TRIPATHY, B. K.; ANURADHA, J. (Ed.). **Internet of Things (IoT)**: technologies, applications, challenges and solutions. [S.l.]: CRC Press, 2017.

VASSEUR, J. P. *et al.* **A survey of several low power Link layers for IP Smart Objects**. Internet Protocol for Smart Objects (IPSO) Alliance, 2010.

VIEIRA, T. P. F. **Gerenciamento Inteligente de Falhas Aplicado a Rede de Sensores Sem Fio Com Integração à Internet das Coisas**. 2017. Dissertação (Mestrado) - Curso de Modelagem Matemática Computacional, Centro Federal de Educação Tecnológica de Minas Gerais, Belo Horizonte, 2017.

WALKER, B. Every day big data statistics—2.5 quintillion bytes of data created daily. **VCloudNews**, [s.l.]. v. 5, abr. 2015.

WEISER, M.; BROWN, J. S. Less intrusion is more useful. **Communications**, Framingham - Massachusetts, v. 31, n. 7, 17 fev. 1997.

WHITE, G. B.; FISCH, E. A.; POOCH, U. W. Computer system and network security. **CRC press**, 2017.

YANG, S. Principle of wireless sensor networks. In: YANG, S. **Wireless Sensor Networks**. London: Springer, 2014. p. 7-47.

YANG, Z. *et al.* Study and application on the architecture and key technologies for IOT. In: INTERNATIONAL CONFERENCE ON MULTIMEDIA TECHNOLOGY, [s.l.], 2011. [S.l.]: IEEE [...], 2011. p. 747-751.

ZHANG, Z. *et al.* IoT security: ongoing challenges and research opportunities. In: IEEE INTERNATIONAL CONFERENCE ON SERVICE-ORIENTED COMPUTING AND APPLICATIONS, 7., 2014, [s.l.]. **Anais** [...]. [S.l.]: IEEE, 2014. p. 230-234.

Unidade 2

Sensores, Microcontroladores e Programação em Internet das Coisas

Convite ao estudo

Prezado estudante,

A Internet das Coisas é uma área fascinante que integra as áreas de Programação, Redes e Hardware da Computação, em um resumo bem simplista. A parte de hardware destaca-se pelo uso e presença dos Sistemas Embarcados, componentes-chave da Internet das Coisas, e que são responsáveis por capturar sinais do meio-ambiente por meio de sensores e atuar no meio-ambiente por meio de atuadores. Mas isso não é “nada” sem o software que determina a alma destes componentes, ditando como deverão agir e reagir mediante os dados coletados. Como o próprio termo diz: “Internet” das Coisas – Internet significa conectividade, e para isso, as “coisas”, estes sistemas embarcados, precisam ter recursos de conectividade para interagir e integrar com outros sistemas por meio da Internet. E nessa integração, sistemas embarcados podem se agrupar a outros sistemas computacionais mais complexos para o desempenho de tarefas também complexas.

Para Sistemas Embarcados utilizados em automação e controle, normalmente utiliza-se a Linguagem C para a sua programação, dado o fato de serem sistemas com recursos mais limitados e esta linguagem permite construir programas que acessam diretamente recursos de hardware do sistema. Já para sistemas computacionais mais poderosos, como o seu próprio computador, podemos utilizar uma linguagem mais prática e versátil, como a Linguagem Python. Na integração dos dois mundos, podemos ter aplicações em Python sendo executadas em um computador, por sua vez, interagindo com um sistema embarcado programado em C.

Dessa forma, a presente unidade lhe permitirá conhecer os fundamentos da linguagem de programação Python e seu uso em projetos de Internet das Coisas, de forma que você tenha a capacidade de traduzir a solução de problemas em um código escrito em Python, pensando em um cenário de integração para a Internet das Coisas.

E em se tratando de integração de dispositivos em um cenário de aplicação de Internet das Coisas, pense no seguinte contexto: devemos automatizar o controle de ventiladores presentes em salas de uma dada empresa. Tais

ventiladores devem ser acionados se sensores detectarem a presença de pessoas no ambiente, e o controle da intensidade dos ventiladores deve ser de acordo com a temperatura ambiente detectada.

De início, veremos como utilizar o Arduino para interagir com sensores e atuadores e demais componentes necessários para comunicação e integração de dados em Internet das Coisas. Na sequência, trabalharemos o conceito de aplicação de sensores e atuadores como mecanismos de coleta e ação/reAÇÃO em um cenário de Internet das Coisas. Por fim, veremos como utilizar a Linguagem Python como ferramenta para Internet das Coisas. Em cada parte, vamos observando como desenvolver os sistemas para permitir o controle que desejamos no contexto apresentado.

Desejo a você um ótimo aprendizado e bons estudos em Internet das Coisas!

Seção 1

O Arduino e a IoT

Diálogo aberto

Prezado estudante,

Vamos agora abrir a unidade apresentando uma das mais poderosas ferramentas atualmente disponíveis para projetistas e entusiastas em Internet das Coisas: o Arduino UNO.

Acredite se quiser, mas o Arduino UNO é um minicomputador. Muito mais simples que seu celular smartphone ou notebook pessoal, mas simples o suficiente para possuir os recursos necessários para desempenhar atividades de controle, seja ligar uma simples lâmpada e até mesmo controlar um robô.

Em nosso contexto de aprendizagem a ser trabalhado, você foi incumbido de desenvolver um sistema que irá automatizar o controle de ventiladores conectados em salas de uma dada empresa, o que será feito com base na detecção de pessoas presentes nas salas e na temperatura ambiente.

Todavia, antes de realmente implementar o sistema, você precisa fazer uma Prova de Conceito para demonstrar um sistema funcionando de forma similar, utilizando somente um Arduino UNO, e tendo à disposição alguns itens como LEDs e botões. As provas de conceito são muito utilizadas em projetos de Internet das Coisas para validar, ou melhor, simular a operação, o comportamento de um sistema a ser implementado, mesmo sem necessariamente utilizar, em princípio, os componentes finais.

Dessa forma, como programar o Arduino para “simular” a leitura da presença de pessoas? Como “simular” a temperatura lida da sala? E como programar o Arduino para “simular” o que será o controle da velocidade do ventilador conforme a temperatura da sala?

Para responder a essas questões, veremos detalhes sobre como programar e utilizar os componentes e sinais que o Arduino UNO possui.

De forma a visualizar tanto as montagens e conexões envolvidas, além da interação com o próprio Arduino UNO, faremos uso da ferramenta online TinkerCad, disponibilizada gratuitamente e que permite simular a execução de programas com montagens usando Arduino UNO.

Uma vez que você tenha aprendido a programar e utilizar um microcontrolador, como o Arduino UNO, você terá a base lógica e criativa para desenvolver sistemas capazes de interagir com o meio real e, mais além, conectado!

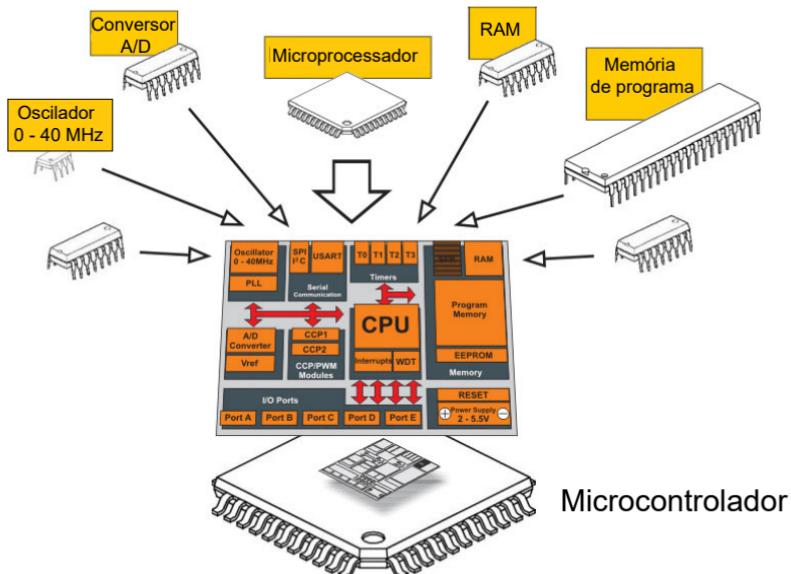
Não pode faltar

A área de Sistemas Embarcados é tão antiga quanto a própria computação. Paralelamente ao surgimento dos computadores voltados para uso científico e aplicações profissionais ou pessoais, também havia diversas empresas desenvolvendo sistemas que seriam utilizados como forma de auxiliar o controle de equipamentos. Tais sistemas eletrônicos embutidos em equipamentos passaram a se chamar **Sistemas Embarcados**, pelo fato de que estão **embarcados** no equipamento, fazendo parte dele e auxiliando no desempenho de suas funções.

O cérebro de Sistemas Embarcados é um componente chamado **microcontrolador**, também apelidado de **computador em um chip**. O microcontrolador recebe este apelido porque ele possui todos os recursos de um computador, tais como CPU (ou microprocessador), RAM, memória de programa, dentre outros, suficiente para operar de forma independente. A Figura 2.1 ilustra um exemplo de componentes presentes em um microcontrolador convencional.

Caso você tente ligar seu computador sem uma unidade de disco-rígido ou sem os pentes de memória RAM, não será possível que o sistema como um todo funcione. Já no caso do microcontrolador, tudo o que é necessário está integrado para funcionar no próprio chip, de forma que basta, literalmente, programar e ligar o microcontrolador para que funcione.

Figura 2.1 | Estrutura de componentes em um microcontrolador



Fonte: <https://bit.ly/2w35d3P>. Acesso em: 3 out. 2019.

Os microcontroladores desempenham um papel fundamental na Internet das Coisas pelo simples fato de que eles são os cérebros das “Coisas”. Existem diversos tipos de microcontroladores disponíveis atualmente no mercado, como mostrado na Figura 2.2. Cada um destes microcontroladores possui arquiteturas distintas, formas de operar distintas. Alguns mais poderosos que outros. Em se tratando de aplicações voltadas à Internet das Coisas, como fazer com que diferentes tipos de sistemas, cada qual com seu microcontrolador, seja capaz de conversar e interagir entre si?

Figura 2.2 | Alguns exemplos de microcontroladores disponíveis no mercado



Fonte: <https://bit.ly/2w7nNb0>. Acesso em: 3 out. 2019.

Conceitos e exemplos de interoperabilidade na IoT

Uma questão de suma importância, em se tratando da interação entre dispositivos conectados, é a **interoperabilidade**, que basicamente trata formas de sistemas diferentes conversarem, interagirem entre si.

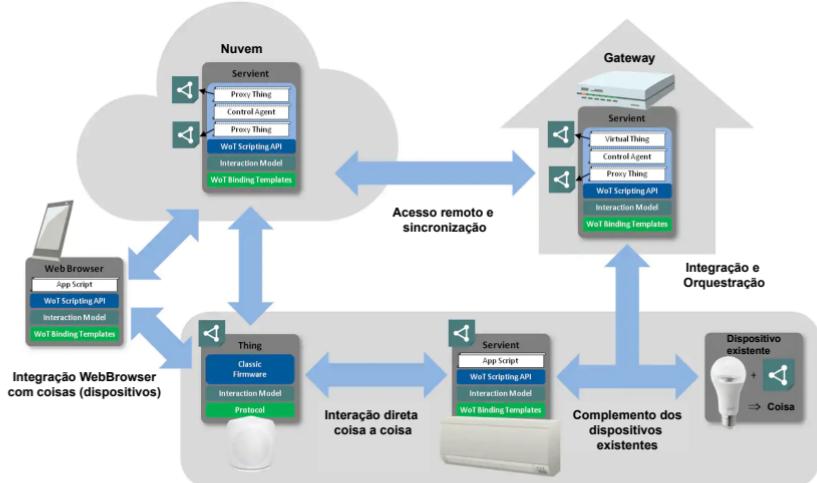
A melhor forma de garantir a interoperabilidade é por meio da padronização de protocolos e mensagens que os dispositivos irão trocar entre si, uma vez conectados em rede. Dessa forma, independentemente de quais sejam os microcontroladores ou sistemas internos de determinados equipamentos, se eles falam uma **língua em comum**, será possível sua interação.

Veja na Figura 2.3 um exemplo de uma arquitetura de componentes interoperáveis em Internet das Coisas. Da nuvem (**cloud**) aos dispositivos (**thing**), mensagens e comandos fluem de um para o outro, exatamente

porque seus sistemas foram implementados para respeitar e adotar um padrão de mensagens em comum. O dispositivo é toda e qualquer coisa que possa ser controlada, tendo sensores e recursos para enviar e/ou receber dados da nuvem.

Esse mesmo padrão é o que também permite o acesso aos recursos por navegadores e aplicações web (**web browser**), por meio da integração de dispositivos em um local comum por meio de **gateways**. Os gateways são **agregadores** – eles são capazes de interagir com dispositivos que utilizem outras formas de comunicação, como WiFi, Bluetooth, ZigBee, dentre outros, e realizar de forma transparente encaminhamento de dados entre uma rede de computadores e tais dispositivos (COELHO, 2017), seguindo, no caso, protocolos para garantir um fluxo de mensagens compatível com toda a infraestrutura integrada, interoperável.

Figura 2.3 | Exemplo de interoperabilidade entre elementos em um sistema de Internet das Coisas



Fonte: <https://www.embarcados.com.br/sucesso-da-iot-interoperabilidade/>. Acesso em: 3 out. 2019.

Em se tratando das mensagens trocadas entre os componentes, um padrão que tem sido amplamente adotado para interoperabilidade é o uso de JSON, que é uma estrutura de dados que organiza informações por meio de uma cadeia de chaves e valores (FLANAGAN, 2014).

Assimile

JSON é a abreviação de *JavaScript Object Notation*, ou, traduzido, Notação de Objeto JavaScript (FLANAGAN, 2014). Como mencionado, é uma estrutura de dados caracterizado em uma cadeia de chaves e seus valores em formato texto. O fato de ser uma estrutura de dados em formato “texto” permite sua ampla adoção por toda sorte de dispositivos e sistemas. Veja, por exemplo, como seria o perfil de mensagem a ser enviada por um dispositivo conectado monitorando temperatura, velocidade de um ventilador e presença em uma dada sala:

```
{"temperatura":27.23, "velocidade":70%, "presenca":1}
```

Qualquer dispositivo que receber este dado será capaz de identificar cada campo e tomar a ação necessária.

O papel do Arduino na IoT

O Arduino é tanto uma família de microcontroladores, que corresponde às placas de desenvolvimento, como uma plataforma de desenvolvimento, que corresponde ao programa Arduino IDE. Ambos surgiram como um esforço para facilitar o acesso e o desenvolvimento com sistemas embarcados, fruto do trabalho do italiano Massimo Banzi e demais colaboradores em meados de 2005.

Antes do Arduino, ferramentas de desenvolvimento para sistemas embarcados eram bem caras, e a programação e o desenvolvimento para tais sistemas dependiam de que o profissional tivesse sólidos conhecimentos tanto de arquitetura de computadores como de Linguagem C, a linguagem de programação mais comumente adotada para Sistemas Embarcados.

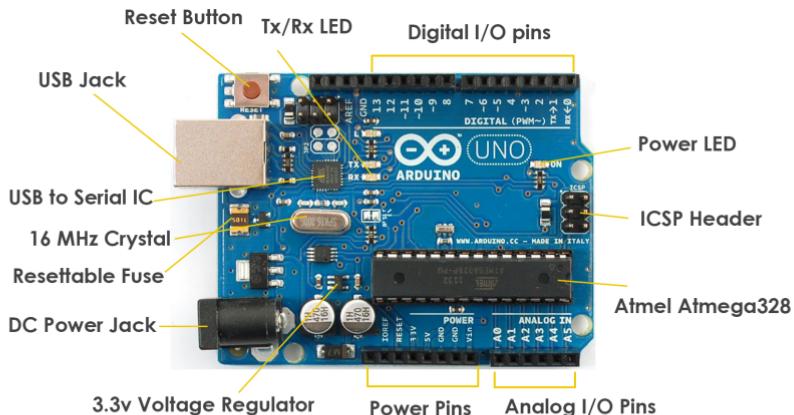
Com o surgimento do Arduino, que é uma ferramenta de mais fácil acesso por ser mais barata, e com o uso do ambiente de desenvolvimento para Arduino, mais e mais pessoas passaram a ter acesso e oportunidade de adentrar ao mundo dos Sistemas Embarcados para o desenvolvimento de produtos, provas de conceitos ou simplesmente por hobby entusiasta.

A plataforma Arduino permite rapidamente prototipar uma montagem integrando sensores, atuadores, módulos de comunicação etc. E o fato de ter uma ampla comunidade de desenvolvedores e um reconhecido suporte da indústria, permite que seja possível encontrar toda sorte de bibliotecas como suporte ao desenvolvimento e interface com componentes utilizados em Internet das Coisas, como sensores, displays, módulos Bluetooth ou WiFi, dentre outros.

E isso facilita muito o desenvolvimento de projetos conectados para Internet das Coisas, pois ao poder contar com bibliotecas pré-existentes para uso de determinado dispositivo significa que você não terá que gastar tempo escrevendo aquela parte do programa que já está pronta.

O Arduino UNO é a placa de desenvolvimento da família Arduino mais comumente adotada e conhecida (OLIVEIRA e ZANETTI, 2015). É caracterizada por ter um microcontrolador ATmega328p operando em tensão de alimentação de 5 V. A placa expõe 14 sinais de GPIO (6 dos quais podem gerar sinais PWM), 6 sinais analógicos, com uma corrente elétrica máxima de saída de 20 mA por sinal GPIO. Esse microncontrolador possui algumas “limitações”, entre elas estão o ATmega328p do Arduino UNO, possui 32 KB de espaço para memória de programa, 2 KB de memória RAM e opera a 16 MHz. Veja na Figura 2.4 o Arduino UNO em detalhes, com legenda de seus componentes.

Figura 2.4 | Placa Arduino UNO



Fonte: <https://www.robomart.com/blog/getting-started-with-arduino-1/>. Acesso em: 3 out. 2019.

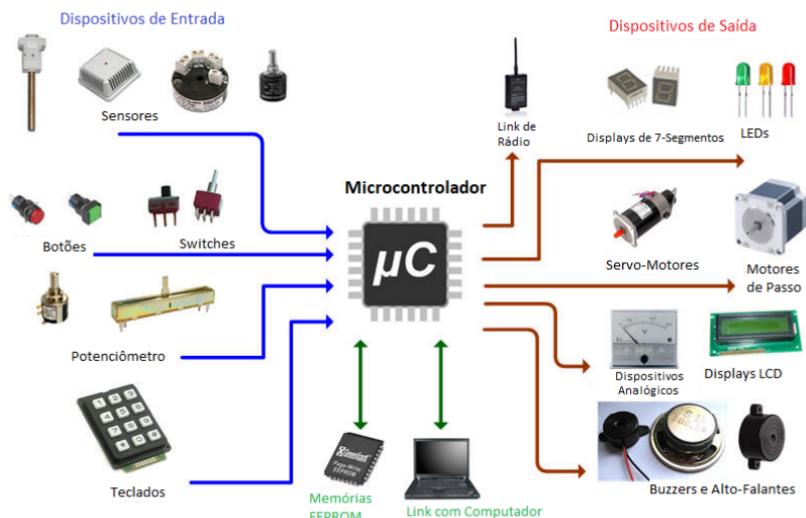
Assimile

GPIO é a abreviação do inglês *General Purpose Input-Ouput*. Trata-se de sinais de lógica digital no microcontrolador que podem assumir, programaticamente, funções de entrada para leitura de sinais, ou de saída para controle de sinais.

Programação do Arduino para a IoT

Em se tratando do uso de Arduino como o cérebro de dispositivos que farão parte de um sistema de Internet das Coisas, é importante considerar o tratamento e a interface com sinais de **entrada** e de **saída** para interagir com o meio-externo. Componentes que farão a interface com sinais de entrada normalmente serão componentes cujo estado o Arduino será capaz de ler, como sensores, botões, chaves mecânicas etc. Componentes que farão a interface com sinais de saída serão componentes a serem controlados pelo Arduino, como LED, motor, buzzer, dentre outros. Veja na Figura 2.5 um exemplo de componentes de entradas e saídas na interface com microcontroladores.

Figura 2.5 | Exemplos de componentes de entradas e saídas para controle com microcontroladores



Fonte: <https://www.petanikode.com/lumpy/>. Acesso em: 14 out. 2019.

O Arduino, com a sua praticidade, acaba sendo um aliado para interface com sensores e atuadores, cujo destaque será apresentado na Seção 2 desta Unidade.

A Linguagem de Programação do Arduino é a Linguagem C, com a observação de que são utilizadas bibliotecas de suporte da Plataforma Arduino com o intuito de facilitar a configuração dos recursos do microcontrolador utilizado.

Refletia

Um programa de computador pode ser inicializado, executar suas rotinas e instruções, exibir algum resultado para o usuário e, por fim, ser finalizado. Todavia, quando tratamos de microcontroladores, há a interação com dispositivos do meio físico, real, de forma que a abordagem de programas a serem executados em microcontroladores precisa ser diferenciada.

Programas em microcontroladores realmente chegam a um fim, ou ficam em constante execução de um laço de repetição **infinito**? Por quê?

Códigos em Linguagem C para a Plataforma Arduino sempre possuirão dois métodos principais: **setup()**, que é executado apenas uma vez quando o microcontrolador é ligado, e é responsável por configurar a operação do sistema; e **loop()**, método que representa o laço infinito responsável por agrupar as ações a serem desempenhadas enquanto o sistema estiver ligado.

Você pode imaginar que a velocidade de operação do Arduino UNO de 16 MHz é pouca coisa, mas não é. O microcontrolador rodará só e somente o programa gravado em sua memória para execução. E a velocidade de execução de instruções no Arduino UNO é tão rápida que precisamos utilizar de rotinas como **delay()** para “segurar” a execução do programa em uma dada quantidade de milissegundos, quando desejado.

Para controlar sinais nas interfaces de **saída** do Arduino, é preciso configurar o sinal GPIO desejado como saída (OUTPUT) por meio do método **pinMode()**. O comando do sinal é desempenhado por meio do método **digitalWrite()**.

Exemplificando

Como exemplo do controle de sinais de saída com Arduino, veja o código em Linguagem C para Arduino apresentado a seguir. Neste código, o sinal 9 é configurado como saída no método **setup()**.

No método **loop()**, esse mesmo sinal é acionado em nível ALTO e nível BAIXO, com uma espera de 1 segundo entre cada comando de forma que seja possível visualizar o LED aceso e apagado por 1 segundo em cada estado. Na Figura 2.6 temos uma montagem, feita pela ferramenta TinkerCAD, na qual um LED vermelho é conectado ao GPIO 9 do Arduino UNO, utilizando uma resistência de 220 Ohm.

```

void setup()
{
    // Configura sinal 9 como saída
    pinMode(9, OUTPUT);
}

void loop()
{
    // Aciona o sinal 9 com nível ALTO (5V)
    digitalWrite(9, HIGH);

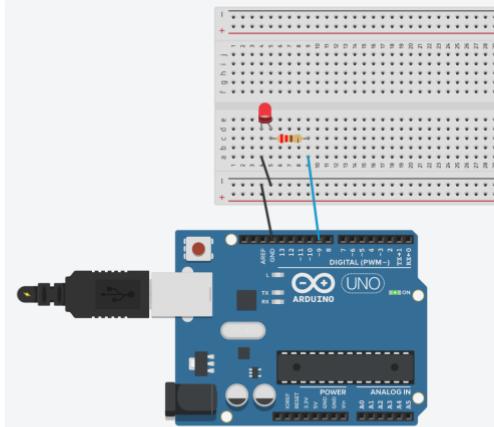
    // Aguarda 1000 ms - 1 segundo
    delay(1000);

    // Aciona o sinal 9 com nível BAIXO (0V)
    digitalWrite(9, LOW);

    // Aguarda 1000 ms - 1 segundo
    delay(1000);
}

```

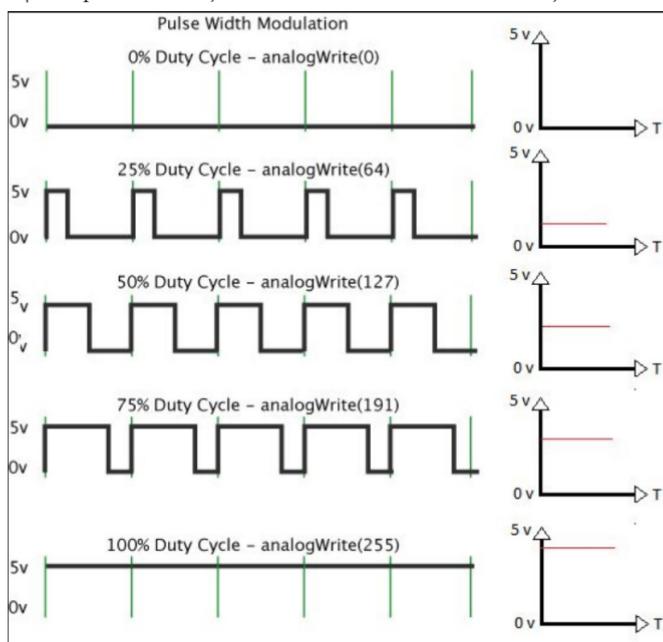
Figura 2.6 | Arduino UNO conectado a um LED



Fonte: elaborado pelo autor.

Os sinais GPIO 3, 5, 6, 9, 10 e 11 (que possuem um símbolo \sim no desenho da placa) do Arduino UNO possuem uma funcionalidade chamada PWM, que é a abreviação de Modulação por Largura de Pulso. Essa funcionalidade permite ir além do típico **Liga/Desliga** lógico desempenhado pelo sinal de saída, de forma que é gerado um sinal periódico com alternância entre quanto tempo o sinal fica em nível **Alto** e quanto tempo fica em nível **Baixo**. A porcentagem de tempo que o sinal fica em nível Alto dentro do período de alternância do sinal é caracterizada por **duty-cycle**. Um sinal com duty-cycle de 100% está 100% em nível Alto, ou seja, totalmente ligado. Já um sinal com duty-cycle de 50% está metade do tempo em Alto, e metade do tempo em Baixo. Como o sinal em nível lógico Alto está em 5 V, e em nível lógico baixo está em 0 V, na média é como se o sinal “estivesse” em 2.5 V, que é o efeito provocado pelo PWM. Veja na Figura 2.7 um exemplo da variação da intensidade do sinal PWM e o efeito provocado. Esse efeito permite controlar a intensidade luminosa de LEDs, movimento de servo-motores, dentre outros recursos.

Figura 2.7 | Exemplo da modulação de sinais PWM e o efeito da modulação



Fonte: <https://i.pinimg.com/736x/ec/79/c9/ec79c97752fb5d8b9250dcfad4f514b0.jpg>. Acesso em: 3 out. 2019.

Para utilizar o recurso de PWM nos sinais de GPIO que suportam tal recurso no Arduino UNO, é preciso utilizar a rotina analogWrite(), informando o sinal de GPIO a ser modulado, e o valor da intensidade desejada, que no caso do Arduino UNO é um intervalo de valores de 0 a 255 para 0 a 100%, respectivamente.

Exemplificando um código que fará um LED aumentar o brilho gradativamente até o brilho máximo, e por fim apagar e começar a aumentar o brilho novamente, veja o trecho de código em Linguagem C destacado adiante.

```
// Variavel global para armazenar a intensidade
```

```
int intensidade = 0;
```

```
void setup()
```

```
{
```

```
// Sinal 9 configurado como Saída
```

```
pinMode(9, OUTPUT);
```

```
}
```

```
void loop()
```

```
{
```

```
// Incrementa valor de intensidade
```

```
intensidade += 5;
```

```
// Valor não pode ser maior que 255
```

```
if (intensidade > 255) {
```

```
    intensidade = 0;
```

```
}
```

```
// Configura sinal PWM de saída
```

```
analogWrite(9, intensidade);
```

```
// Aguarda meio segundo
```

```
delay(500);
```

```
}
```

Para poder realizar a leitura de sinais nas interfaces de **entrada** do Arduino, é preciso configurar o sinal GPIO desejado como entrada (INPUT), também por meio do método **pinMode()**. Para a leitura do estado do sinal de entrada, usa-se o método **digitalRead()**.

A leitura de sinais lógicos é algo muito realizado em toda sorte de sistemas embarcados. Desde seu controle remoto, a máquina de lavar, micro-ondas, pense em todo e qualquer sistema que realize a leitura de botões. Veja no material disponibilizado adiante como realizar a leitura de sinais lógicos, por meio do exemplo da leitura do estado de um botão utilizando Arduino em Linguagem C.



No material apresentado neste QR code, veja como realizar a leitura de sinais lógicos, por meio do exemplo da leitura do estado de um botão utilizando Arduino em Linguagem C.

Como você viu no exemplo apresentado anteriormente, a ligação do botão com o Arduino UNO foi realizada usando uma técnica chamada de Pull-Up. Nesta técnica, usa-se uma resistência de valor elevado, como por exemplo 10K Ohm, conectando o sinal do botão ao sinal de alimentação elétrica positiva (Vcc), de forma que quanto o botão não está pressionado, o Arduino irá ler nível lógico ALTO. Quando o botão estiver pressionado, o Arduino irá ler nível lógico BAIXO.

Pesquise mais

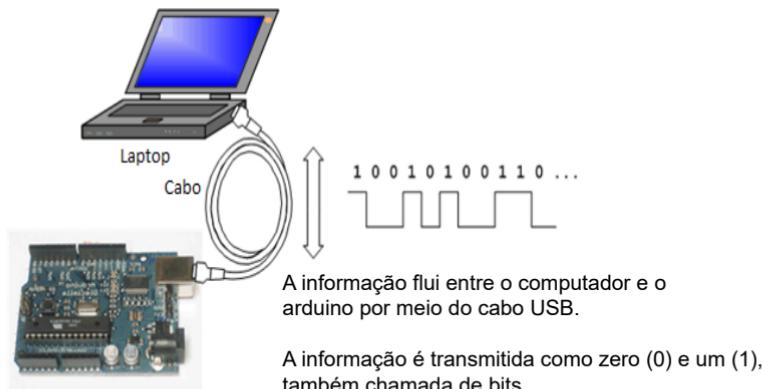
Sinais em Pull-Up e Pull-Down são muito utilizados em sistemas embarcados para garantir níveis lógicos estáveis tanto em nível ALTO quanto em nível BAIXO, respectivamente.

NOGUEIRA, Danilo. **Funcionamento Resistores Pull-Up e Pull-Down**.
Portal AutoCore Robótica, 27 ago. 2018.

Outra parte muito interessante do Arduino UNO é que ele possui um recurso chamado **Comunicação Serial**. Basicamente, o microcontrolador pode enviar e receber bytes de dados de forma serial/sequencial em uma velocidade de comunicação configurada em **bits-por-segundo**, normalmente abreviado por **bps**. Dentre as velocidades comumente configuradas

no Arduino UNO estão 9600 bps, que consiste em 9600 bits-por-segundo, e 115200 bps, que consiste em 115200 bits-por-segundo. Na literatura técnica de sistemas embarcados, essa velocidade de comunicação serial também é chamada **baudrate**. Esse recurso de comunicação permite ao Arduino UNO tanto comunicar-se com outras placas como também comunicar-se com um computador, como mostrado na Figura 2.8.

Figura 2.8 | Comunicação serial entre Arduino e computador



Fonte: <https://www.ladyada.net/learn/arduino/lesson4.html>. Acesso em: 3 out. 2019.]

Para fazer uso da Comunicação Serial do Arduino UNO é preciso configurar a Porta Serial do microcontrolador por meio do método **Serial.begin()**, informando a velocidade de comunicação desejada, sendo chamado dentro de **setup()**. Uma vez configurada a comunicação serial do microcontrolador, para que o Arduino possa enviar mensagens ao computador basta chamar o método **Serial.println()** com o texto desejado ou variável cujo valor deseja ser informado .



Para ilustrar tanto a configuração da Porta Serial como o a exibição de mensagens pela Serial, veja o material disponibilizado neste QR code.

Pesquisa

A comunicação Serial com o Arduino permite não só enviar mensagens ou dados para um computador, mas também permite receber dados

e mensagens! Dessa forma, é possível que o computador comande o Arduino para desempenhar uma dada função, como o ligar/desligar de um LED, por exemplo.

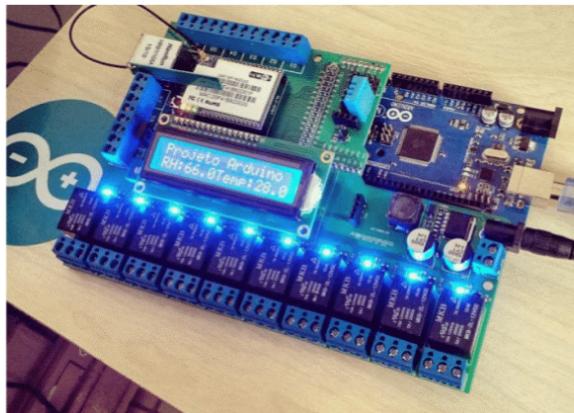
SOUZA, Fabio. **Arduino - Comunicação Serial**. Portal Embarcados, 19 dez. 2017.

Construindo um sistema real com o Arduino

Em se tratando do ambiente TinkerCad, visualizamos montagens que envolvem o uso de placas de prototipação, componentes e fios conectando o Arduino a estes componentes na placa de prototipação. Todavia, um sistema real precisa possuir um arranjo fixo e estável já com os componentes que serão utilizados montados em uma placa de circuito impresso projetada para a aplicação desejada.

Veja o caso da Figura 2.9, na qual se vê uma placa de automação residencial que possui display LCD, sensor de temperatura e umidade, módulo de comunicação WiFi e um arranjo de sinais de entrada e sinais de saída por meio de relés para acionamento de cargas. O único fio “solto” é a antena do módulo de comunicação. Os demais componentes estão soldados na placa, que por sua vez é afixada em uma placa Arduino MEGA (outra variante da família Arduino).

Figura 2.9 | Placa de automação residencial com Arduino



Fonte: <https://www.embarcados.com.br/shield-automacao-e-alarme-residencial-v5/>. Acesso em: 3 out. 2019.

O Arduino MEGA, neste caso, deverá realizar a configuração dos seus sinais de entrada e saída para corresponder com as ligações esperadas na placa de controle para automação residencial.

E uma vez programado e configurado o Arduino, será possível realizar a leitura de sinais, exibição de mensagens no display LCD, controle dos relés, dentre outros recursos que a placa oferece, neste caso.

Agora você viu como programar o Arduino IDE para realizar o controle de sinais lógicos de saída, além de modular sinais por meio de PWM. Você viu também como realizar a leitura de sinais lógicos e a comunicação com um computador por meio da Porta Serial do Arduino. Dessa forma, você já é capaz de criar programas que consigam desempenhar atividades de controle com base em eventos externos, além de reportar o estado do equipamento a um computador conectado!

Sem medo de errar

Você foi selecionado para realizar a prova de conceito de que o Arduino será capaz de interagir com sensores de presença e de temperatura para controlar ventiladores presentes em salas de uma empresa. Os ventiladores só devem ser acionados caso seja detectada a presença de pessoas no local. E uma vez ligados, os ventiladores funcionarão de forma mais ou menos forte de acordo com a temperatura lida. Ou seja, em um dia mais quente, os ventiladores ficarão ligados com uma velocidade maior quando houver pessoas na sala!

Para simular esse tipo de sistema em uma montagem com Arduino UNO disporo somente de LEDs e botões, você pode:

- Usar um botão como sinal para simular a presença de pessoas, agindo como o “sensor” de presença.
- Usar outro botão como sinal para simular o sensor de temperatura, de forma que cada vez que o botão for pressionado, o valor da temperatura irá subir até um determinado valor de referência (40°C , por exemplo). Após subir até o valor de referência, o sensor “simulado” pode voltar a incrementar a temperatura a partir de 0°C ou outro valor de temperatura “base”.
- Usar um LED com intensidade de brilho controlada via PWM, representando a intensidade de controle do ventilador.

Considerando que o botão para simular a presença de pessoas esteja conectado no GPIO 2, o botão para simular o sensor de temperatura esteja no GPIO 3

e o LED esteja no GPIO 11 (que suporta PWM), podemos arquitetar a Prova de Conceito com base no seguinte código em Linguagem C para Arduino UNO. Na Figura 2.10 vemos a montagem do setup na ferramenta TinkerCad.

```
// Variavel global para leitura de presenca  
int leitura_presenca = 0;  
  
// Variavel global para leitura de temperatura  
int leitura_temperatura = 0;  
  
// Variavel global para armazenar valor de temperatura  
int temperatura = 0;  
  
// Variavel global para controle do ventilador  
int intensidade_ventilador = 0;  
  
  
void setup()  
{  
    // Configura sinal 2 como entrada (presenca)  
    pinMode(2, INPUT);  
    // Configura sinal 3 como entrada (temperatura)  
    pinMode(3, INPUT);  
    // Configura sinal 11 (PWM) como saída (LED)  
    pinMode(11, OUTPUT);  
}  
  
  
void loop()  
{  
    // Realizamos leitura do nivel logico do sinal 2 (presenca)  
    leitura_presenca = digitalRead(2);  
    // Realizamos leitura do nivel logico do sinal 3 (temperatura)  
    leitura_temperatura = digitalRead(3);  
    if (leitura_temperatura == 0) {
```

```
// Incrementa intensidade da temperatura
temperatura += 5;

// Se passar de 40 graus, comeca a incrementar novamente
if (temperatura > 40) {
    temperatura = 0;
}
```

```
// Leitura em nivel BAIXO - Botao Pressionado
if (leitura_presenca == 0) {
    // A intensidade do ventilador ira virar uma escala
    // de 0 a 255 com base na medida do sensor de temperatura,
    // que por sua vez mede de 0 a 40 graus, no nosso caso.
    intensidade_ventilador = map(temperatura, 0, 40, 0, 255);
    analogWrite(11, intensidade_ventilador);
}

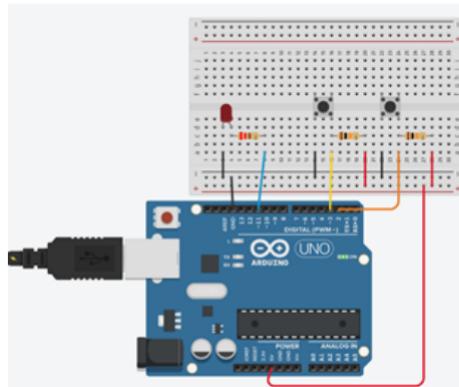
else {
    // Sem pessoas na sala, ventilador desligado
    digitalWrite(11, LOW);
}

// Delay de 1 segundo ate proxima rodada
```

```
delay(1000);
```

```
}
```

Figura 2.10 | Exemplo de montagem com Arduino tem TinkerCad para Prova de Conceito



Fonte: elaborada pelo autor.

Avançando na prática

Exibindo relatórios de informações via Serial

Na solução original é possível ver o sistema funcionando sob um aspecto visual, no qual o LED gradativamente brilhará mais e mais forte conforme o botão de temperatura é pressionado. Porém, não é possível saber o valor da temperatura lida pelo sensor “simulado”, nem o valor da intensidade que está sendo tomado como referência para o controle do LED, que está no lugar do ventilador nessa Prova de Conceito. Como saber, então, os valores lidos e utilizados na operação do programa de controle gravado no Arduino UNO?

Resolução da situação-problema

Para saber o que se passa no microcontrolador do Arduino UNO, podemos fazer uso da Comunicação Serial. Dessa forma, poderemos enviar a um computador conectado ao Arduino UNO os valores do sensor de temperatura e de controle de intensidade do LED.

Considerando que a Comunicação Serial seja configurada em **setup()** como **Serial.begin(9600)**, podemos proceder com a adição das chamadas **Serial.print()**

e `Serial.println()` no método `loop()` do programa em Linguagem C em execução no Arduino UNO, como destacado em negrito no trecho de código apresentado adiante:

```
void loop()
{
    // Realizamos leitura do nível logico do sinal 2 (presença)
    leitura_presença = digitalRead(2);

    // Realizamos leitura do nível logico do sinal 3 (temperatura)
    leitura_temperatura = digitalRead(3);

    if (leitura_temperatura == 0) {
        // Incrementa intensidade da temperatura
        temperatura += 5;

        // Se passar de 40 graus, começa a incrementar novamente
        if (temperatura > 40) {
            temperatura = 0;
        }
    }

    Serial.print("Temperatura: ");
    Serial.println(temperatura);

    // Leitura em nível BAIXO - Botão Pressionado
    if (leitura_presença == 0) {
        // A intensidade do ventilador irá virar uma escala
        // de 0 a 255 com base na medida do sensor de temperatura,
        // que por sua vez mede de 0 a 40 graus, no nosso caso.
        intensidade_ventilador = map(temperatura, 0, 40, 0, 255);
        analogWrite(11, intensidade_ventilador);

        Serial.print("Ventilador ligado com intensidade: ");
    }
}
```

```

Serial.println(intensidade_ventilador);

}

else {
    // Sem pessoas na sala, ventilador desligado
    digitalWrite(11, LOW);
    Serial.println("Ventilador desligado...");

}

// Delay de 1 segundo ate proxima rodada
delay(1000);
}

```

Uma vez gravado e colocado em execução, o programa exibirá os valores desejados no Monitor Serial, como mostrado na Figura 2.11.

Figura 2.11 | Exibição dos valores desejados no Monitor Serial para acompanhamento da execução do programa



The screenshot shows the Arduino Serial Monitor window titled "Monitor serial". The text area displays the following output:

```

Monitor serial
Temperatura: 25
Ventilador desligado...
Temperatura: 25
Ventilador ligado com intensidade: 159
Temperatura: 25
Ventilador ligado com intensidade: 159
Temperatura: 25
Ventilador desligado...

```

Below the text area are three buttons: "Env.", "Apag.", and a button with a double arrow icon.

Fonte: elaborada pelo autor.

Faça valer a pena

1. Para controle de um determinado componente, a especificação do fabricante determina que será acionado quando um sinal com tensão de 4.5 V for aplicado em sua entrada.

Considerando que um Arduino UNO seja utilizado para acionar o componente em questão, é correto assinalar o seguinte valor de duty-cycle a ser empregado na chamada analogWrite, considerando o uso de GPIO com PWM:

- a. 230.

- b. 231.
- c. 220.
- d. 255.
- e. 200.

2. Um Arduino UNO está conectado a um computador por meio da conexão da Porta USB, sendo a mesma utilizada para realizar a Comunicação Serial com o microcontrolador. Este Arduino UNO realiza a leitura periódica de sensores e estados de acionamento de relés presentes numa placa de controle, também conectada ao Arduino. A quantidade de dados lidos é reportada ao computador em um total de 140 bytes pela Comunicação Serial.

É correto assinalar que cada transferência de dados entre o Arduino UNO e o computador, via Comunicação Serial em 19200 bps, ocorrerá no seguinte intervalo de tempo:

- a. 1 segundo.
- b. 100 milissegundos.
- c. 500 milissegundos.
- d. 58 milissegundos.
- e. 47 milissegundos.

3. Uma máquina de venda de refrigerantes utiliza um painel com botões para que o usuário selecione qual o refrigerante desejado. Um projetista que está adaptando a máquina para ser controlada via Arduino UNO precisa que o sinal a ser lido do botão seja computado apenas uma vez, de forma a coordenar demais elementos do equipamento para dispensar um só refrigerante por vez.

Este projetista começou a fazer um esboço de código para testar o funcionamento do sistema, conectando a placa Arduino UNO ao seu computador. Até o momento, está executando o sistema com o seguinte código-fonte em Linguagem C, monitorando apenas um botão do painel, ligado na GPIO 2 do Arduino UNO:

```
int estado_botao = 0;  
int leitura_botao = 0;
```

```
void setup()
```

```

{

// Configura sinal 2 como entrada (presenca)
pinMode(2, INPUT);
Serial.begin(9600);

}

void loop()
{
    leitura_botao = digitalRead(2);
    if (leitura_botao != estado_botao) {
        Serial.println("Botao pressionado!");
    }

    estado_botao = leitura_botao;
    delay(1000);
}

```

Com base no que foi observado e estudado em se tratando de programação em Linguagem C para Arduino UNO e simulação em ambiente TinkerCad, é correto dizer que, ao pressionar o botão do painel conectado na GPIO 2 do Arduino UNO, a mensagem “Botao pressionado!” irá aparecer da seguinte forma no Monitor Serial:

- A mensagem irá aparecer quando o botão for pressionado e quando for liberado.
- A mensagem irá aparecer somente quando o botão for pressionado.
- A mensagem irá aparecer somente quando o botão for liberado.
- A mensagem não irá aparecer.
- A mensagem ficará aparecendo enquanto o botão permanecer pressionado.

Seção 2

Redes de sensores

Diálogo aberto

Prezado estudante, vamos agora adentrar um pouco mais nos recursos que permitem aos Sistemas Embarcados realizar a interface com o meio físico e real.

Assim como seus sentidos lhe permitem ver, ouvir, sentir, se equilibrar, temos os recursos dos sensores que permitem aos sistemas embarcados captar ampla variedade de sinais do meio externo para que sejam então processados pelas rotinas de seus programas em execução.

E assim como você consegue interagir de volta com o meio ambiente ao seu redor, temos também os atuadores, sistemas e mecanismos capazes de realizar alguma ação no meio externo ao sistema embarcado que caracteriza um elemento num sistema da Internet das Coisas.

Mas pense bem: seus sentidos seriam de alguma utilidade sem seu cérebro? Da mesma forma sensores e atuadores dependem de estar conectados a sistemas computacionais para sua devida interação, como exemplo do Arduino, que é o nosso cérebro em questão, pois é quem irá realizar a leitura de sensores, processar as rotinas de programa e controlar os atuadores.

Para a aplicação dos conceitos de sensores e atuadores no âmbito da Internet das Coisas, vamos considerar um Projeto que consista em um sistema integrado para detectar a presença de pessoas numa sala para o controle programado de um ventilador. Esse sistema é composto, em princípio, por um sensor de presença e um atuador responsável por controlar um ventilador, ambos conectados à internet. De acordo com a presença ou não de pessoas no ambiente, o sensor de presença notificará o atuador, que por sua vez deverá ligar ou desligar o ventilador da sala com base nas notificações recebidas pelo sensor de presença.

E então, como realizar essa interação entre o sensor e o atuador conectados à internet? Como programar o Arduino para atuar como o cérebro capaz de realizar de um lado a leitura de sinais e de outro o controle de atuadores?

Como o objetivo é ter sensores e atuadores conectados à internet, inicialmente é preciso saber quais ferramentas ou módulos de suporte precisamos integrar com o Arduino UNO para a realização da troca de dados e comandos pela internet.

Na sequência, para utilizar de sensores e atuadores em sistemas embarcados, é preciso então saber como tais componentes operam na interface com sistemas digitais microcontrolados, como o Arduino, de forma a apropriadamente interagir com eles – afinal de contas, trata-se de um trabalho em equipe!

Abrindo mais ainda o leque sobre o trabalho em equipe, devemos, então, saber como montar e utilizar módulos de conectividade para permitir a troca de dados desejada entre sensores e atuadores.

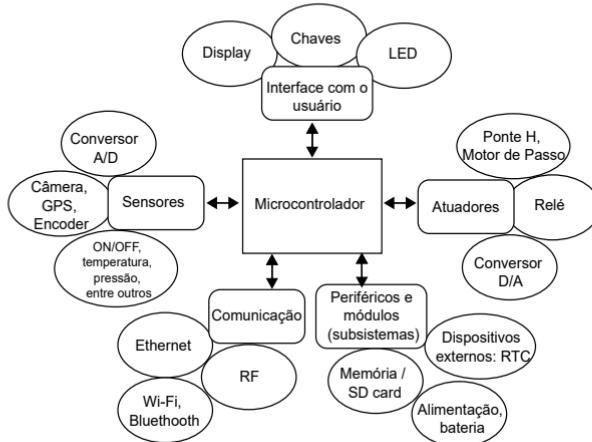
Depois de muita conversa, mãos à obra e ótimos estudos!

Não pode faltar

Em se tratando de Internet das Coisas, a fronteira com o meio-ambiente está nos sensores, para captar diversos tipos de sinais pela tradução de grandezas físicas diversas (temperatura, luminosidade, umidade etc.) em uma intensidade elétrica proporcional. E a forma de o dispositivo reagir está nos atuadores, que fazem o outro lado do caminho, traduzindo sinais elétricos em alguma ação física no meio externo, seja movimentando um motor ou acionando uma chave, contato ou mesmo piscando um simples LED (THOMAZINI; ALBUQUERQUE, 2011).

Entre o sensor e o atuador, temos o microcontrolador, como o Arduino visto na seção anterior, que atua como cérebro do sistema, processando os sinais lidos dos sensores e controlando os atuadores com base em suas instruções de programa. Vamos pensar em como seria um sistema embarcado qualquer de uma forma mais generalista, possuindo sensores para entrada de dados, atuadores para agir no meio-ambiente, além de interfaces com outros periféricos que permitem ao mesmo ter comunicação com outros dispositivos, armazenar dados, alimentação elétrica, além de interagir com o usuário, como mostrado no diagrama da Figura 2.12.

Figura 2.12 | Diagrama geral de componentes presentes em um sistema embarcado com microcontrolador



Fonte: <https://www.embarcados.com.br/sistemas-embarcados-e-microcontroladores/>. Acesso em: 22 set. 2019.

Veja que no esquema mostrado na Figura 2.12 há um bloco muito importante chamado **comunicação**, que engloba os possíveis componentes capazes de permitir ao equipamento interagir e comunicar com uma rede de computadores, com a internet e inclusive com outros equipamentos. Em se tratando de comunicação, há um módulo que merece destaque, o **ESP8266**, capaz de conectar o sistema embarcado a uma rede Wi-Fi, de forma que o sistema poderá se comunicar com outros dispositivos em rede e com a internet, caso a rede tenha acesso à internet.

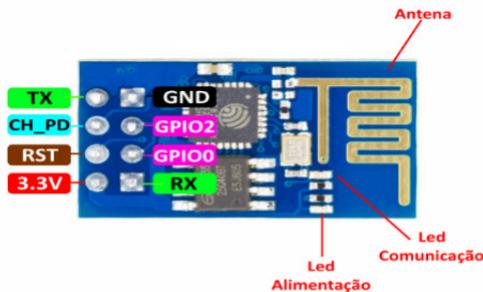
Pesque mais

O ESP8266 é controlado pela Serial – comunicação que ocorre pela transmissão sequencial de dados por meio dos sinais TX e RX, por uma série de comandos chamados **Comandos AT**, um padrão adotado na indústria para comandos que começam com as letras AT. Conhecendo os comandos suportados pelo ESP8266 é possível comandá-lo para se conectar a uma rede Wi-Fi e até mesmo exibir uma página Web quando requisitado.

- THOMSEN, A. *Tutorial Módulo Wireless ESP8266 com Arduino*. FILIPEFLOP, 23 jun. 2015.

Na Figura 2.13 é exibida uma versão do ESP8266 chamada ESP-01, com destaque para legenda de seus sinais e componentes presentes no módulo.

Figura 2.13 | Módulo ESP8266 e legenda dos seus sinais e componentes



Fonte: <https://www.filipeflop.com/produto/modulo-wifi-esp8266-esp-01/>. Acesso em: 26 set. 2019.

Com relação aos sinais presentes no ESP8266 destacados na Figura 2.12, podemos descrever:

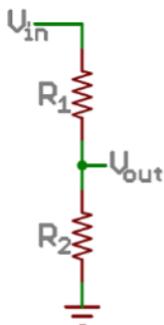
- TX – Sinal para transmissão de dados via Serial, conectado no sinal RX da outra parte.
- RX – Sinal para recepção de dados via Serial, conectado no sinal TX da outra parte.
- RST – Sinal de Reset do módulo, reinicia o módulo em nível lógico Baixo (0).
- 3.3 V – Sinal de alimentação positiva do módulo, em 3.3 V.
- GND – Sinal de terra do módulo.
- CH_PD – Sinal de Chip Power Down, ativa hibernação em nível lógico Baixo (0).
- GPIO2 – Sinal de controle de GPIO de número 2.
- GPIO0 – Sinal de controle de GPIO de número 0.

Para conectar o ESP8266 com o Arduino UNO é preciso tomar cuidado com o seguinte fato: O ESP8266 é um componente que funciona em 3.3 V, e o Arduino Uno funciona em 5 V. Para contornar esse problema, fazemos uso de divisores de tensão para equilibrar as tensões de operação entre as partes. Um exemplo tradicional de divisor de tensão é mostrado na Figura 2.14. A

fórmula que descreve como ficará a tensão de saída é, então, caracterizada da seguinte forma:

$$V_{out} = V_{in} \cdot \frac{R_2}{R_1 + R_2}$$

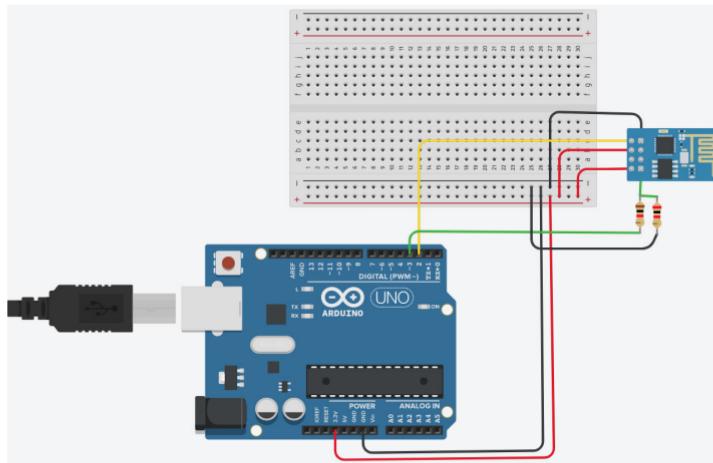
Figura 2.14 | Exemplo de circuito para divisor de tensão



Fonte: <https://learn.sparkfun.com/tutorials/voltage-dividers/all>. Acesso em: 30 set. 2019.

Com base nessa adaptação de sinal, é possível realizar com segurança a interface entre o Arduino e o ESP8266 para trabalhar em conjunto, como mostrado no arranjo desenvolvido com TinkerCad na Figura 2.15.

Figura 2.15 | Exemplo de interface entre Arduino e ESP8266 no programa TinkerCad



Fonte: elaborada pelo autor.

Pesquise mais

O [TinkerCad](#), da Autodesk, é uma plataforma online que permite a montagem e até mesmo a programação e simulação de sistemas montados com Arduino UNO. O usuário pode realizar gratuitamente seu cadastro na plataforma e começar a criar e simular componentes online. É possível simular diversos tipos de sensores e componentes.

Observe o par de resistores próximos ao sinal de RX do ESP8266 na Figura 2.15 – eles estão fazendo o papel do divisor de tensão comentado. Observe também que o ESP8266 está sendo alimentado pelo sinal de 3.3V do Arduino. O setup mostrado na Figura 2.14 é suficiente para já proporcionar ao Arduino a habilidade de comunicar com o módulo ESP8266, e assim ter conectividade com rede local ou internet.

Assmile

Uma vez que o arranjo do Arduino com o ESP8266 esteja conectado em uma rede WiFi, haverá um endereço IP associado à conexão do dispositivo na rede, como por exemplo “192.168.1.20”, e é por meio desse endereço IP que conseguiremos interagir com o sistema, enviando e recebendo dados do mesmo, conforme programado no Arduino em sua interação com o ESP8266 e demais componentes, como sensores e atuadores.

Configuração e leitura de sensores via Internet

Considerando um sistema microcontrolado com um módulo capaz de conectar a uma rede com Internet, é possível realizar a leitura de dados de sensores e assim transmitir os dados lidos pela rede. No nosso caso, o Arduino poderá ler dados de sensores e transmiti-los por meio do ESP8266, bastando programar no Arduino como será feita a leitura do sensor e a interação com o ESP8266 para transmitir os dados para outro dispositivo local ou na internet.

Observação: O dispositivo ESP8266 está disponível na ferramenta TinkerCad, porém não é possível simular por meio da ferramenta. Nos exemplos que serão listados na presente seção teremos algumas funções que **simbolicamente** destacarão pontos no código onde devem ser realizados processos de configuração, leitura e escrita de dados no ESP8266 para comunicação via rede, a saber:

- [ConfiguraEsp8266\(\)](#) – Configura o ESP8266 para conectar e operar em uma rede WiFi.

- EnviaDadoEsp8266(dado) – Configura o ESP8266 para enviar o dado para um outro dispositivo na rede local ou internet.
- Dado = recebeDadoEsp8266() – Requisita ao ESP8266 a leitura de algum dado que tenha chegado da rede local ou da internet.

Uma aplicação cada vez mais comum é a integração de sensores de presença conectados. Isso permite a empresas a gestão de salas de reuniões, e até mesmo o monitoramento inteligente para alarmes residenciais, por exemplo .



Veja mais detalhes sobre a integração do Arduino UNO com um sensor de presença e o módulo ESP8266 neste QR code.

Sensores como o PIR são sensores que operam respondendo sinais lógicos ALTO (1) e BAIXO (0), que um microcontrolador como o Arduino facilmente é capaz de interpretar. Mas nem todos os sensores respondem dessa forma: temos sensores que respondem de forma **analógica**. Isso quer dizer que o sinal que o sensor irá responder não será nem ALTO, nem BAIXO, mas uma variação de sinal elétrico dentro de uma faixa de valores que corresponde à intensidade da grandeza física medida pelo sensor.

Pesquise mais

Conversão analógico-digital é uma parte importante da interface com sensores que respondem de forma analógica. E o entendimento e aplicação apropriados desses conceitos permitem a elaboração de aplicações capazes de computar apropriadamente os sinais fornecidos.
INSTITUTO NEWTON C. BRAGA. **Como funcionam os Conversores A/D - parte 1 (ART224)**. [s.d.]

Para lidar com sensores analógicos é preciso fazer uso de sistemas embarcados que tenham **conversores analógico-digitais**, que farão a interpretação analógica dos níveis de tensão fornecidos pelo sensor em informações digitais capazes de serem trabalhadas pelo software do sistema embarcado.

Assimile

É importante saber a equivalência entre o valor numérico lido pelo conversor analógico-digital com relação à tensão elétrica que está sendo efetivamente lida pelo conversor, pois os sensores normalmente operam com um chamado **fator de escala**, que relaciona a grandeza física medida com a tensão elétrica que ele responde, como por exemplo 20 mV / °C, para um dado sensor de temperatura.

O Arduino UNO possui um conversor analógico-digital com resolução de 10-bits, operando com uma tensão de referência de 5 V. Isso significa que o Arduino UNO é capaz de interpretar sinais analógicos que variem de 0 a 5 V dentro de uma correspondente faixa de valores inteiros que vão de 0 a 1023 (10 bits de valores). O processo de conversão ocorre por amostras que são tiradas do sinal analógico ao longo do tempo.

No TinkerCad temos o sensor TMP36, que é um sensor analógico de temperatura que responde com um fator de escala de 10 mV / °C medido. Ele apresenta um **offset** de 500 mV no sinal analógico de temperatura, o que significa que é preciso subtrair 500 mV da tensão elétrica de medida, e assim computar a temperatura com base no fator de escala .



Veja como realizar a interface do sensor TMP36 com Arduino via Tinker-Cad no material disponível neste QR code.

Configuração e interações com atuadores via Internet

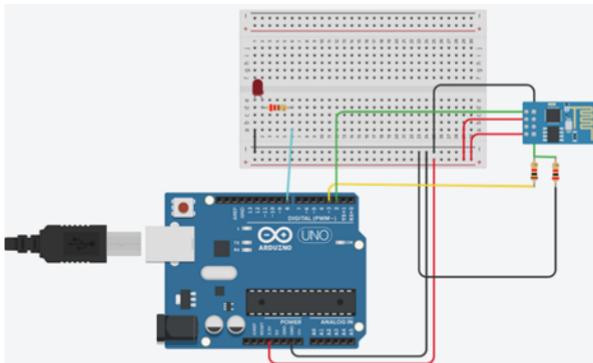
Da mesma forma que sensores conectados podem enviar dados pela rede local ou internet, é possível trabalhar também no sentido contrário: receber comandos pela rede local ou internet para o controle de algum dado componente, um atuador.

Refita

Um LED convencional pode consumir cerca de 20 mA de corrente elétrica. Quanto mais LEDs forem conectados ao Sistema Embocado, maior será o consumo. Considerando que a corrente elétrica total que um Arduino consegue fornecer em todos os seus sinais de saída é de 200 mA, poderíamos ligar até cerca de uns 10 LEDs no total. E caso queira-se controlar mais LEDs, como fazer?

Vamos considerar o exemplo de um LED controlado remotamente. Neste caso, sendo um atuador, espera-se que venha um comando remoto, de forma que devemos consultar o ESP8266 se algum dado chegou pela rede e assim orientar o fluxo de controle do programa sob o atuador em questão. Nesse caso, considere a montagem apresentada na Figura 2.16, realizada no TinkerCad, onde um Arduino UNO está conectado a um ESP8266 para interface com a rede, e também está conectado a um LED, o qual será remotamente controlado.

Figura 2.16 | LED conectado à rede com ESP8266 e Arduino no TinkerCad



Fonte: elaborada pelo autor.

Exemplificando

No loop() o ESP8266 é consultado se há algum dado novo, de forma que a variável dado é populada com o valor do comando remoto. Este valor é usado como referência para ligar ou não o LED conectado à placa. É um exemplo que embasa como é o princípio de um comando remoto, de um atuador conectado à internet.

Código em Linguagem C para Arduino – Interface com LED e ESP8266 com Arduino UNO

```
#include <SoftwareSerial.h>

// Cria uma comunicacao Serial via Software no Arduino - Sinal RX no
// pino 2, e TX no pino 3
SoftwareSerial esp8266(2, 3);

void setup()
```

```

{
    pinMode(8, OUTPUT); // sinal do LED como saída
    digitalWrite(8, LOW); // LED inicia apagado
    Serial.begin(9600); // Configura serial do Arduino com ESP8266
    esp8266.begin(115200);
    // Chamada de rotina para configurar o ESP8266
    // deverá ser criada e escrita com base nos comandos AT necessários!
    configuraEsp8266();
    // Espera 1 segundo
    delay(1000);
}

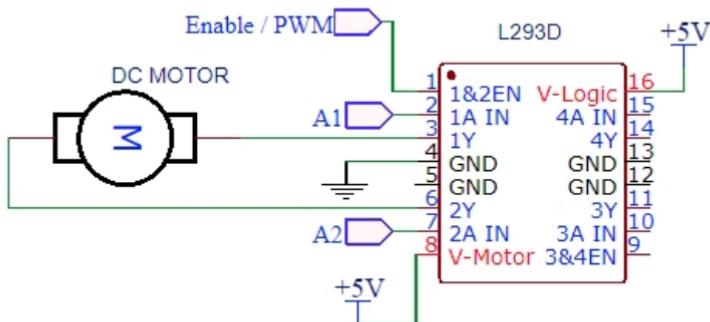
void loop() {
    // Recebe dado da rede pelo esp8266
    int dado = recebeDadoEsp8266();
    // Verifica se dado eh 1 ou 0 para ligar ou desligar o motor
    if (dado == 1) {
        digitalWrite(8, HIGH); // liga LED
    } else if (dado == 0) {
        digitalWrite(8, LOW); // desliga LED
    }
    // aguarda 1 segundo até verificar novamente
    delay(1000);
}

```

Além de LEDs, é possível controlar elementos que consomem ainda mais corrente elétrica, como motores de passo e motores de corrente contínua (DC), fazendo uso da Ponte H, como exemplo da L293D, disponível no TinkerCad.

Para controlar elementos que consomem mais corrente elétrica é preciso fazer uso de **drivers**, que são circuitos elétricos de potência, capazes de coordenar a operação de componentes que funcionam com maiores valores de tensão e corrente elétrica do que sistemas digitais normalmente conseguem coordenar. Um exemplo bem comum de **driver** é a chamada Ponte-H, que consiste em um arranjo de transistores que possibilitam o controle de motores, sendo um componente muito utilizado em robótica, por exemplo. Uma Ponte-H é capaz de controlar o movimento de motores elétricos em sentido horário ou anti-horário, além de ser também capaz de coordenar a intensidade do movimento do motor. Na Figura 2.17 temos um esquema de ligação de um motor elétrico com a Ponte-H L293D.

Figura 2.17 | Esquema de ligação para 1 motor com Ponte-H L293D



Fonte: <https://protosupplies.com/wp-content/uploads/2018/03/L293D-Schematic.jpg>. Acesso em: 26 set. 2019.

Ao utilizar um **driver** conectado ao seu sistema embarcado conectado, agora é possível controlar elementos diversos, tais como motores elétricos que podem orientar o movimento de um robô ou simplesmente ligar ou desligar um ventilador elétrico, por exemplo.

Veja como realizar a interface do Arduino UNO com uma Ponte H e o módulo ESP8266 no material disponibilizado adiante.

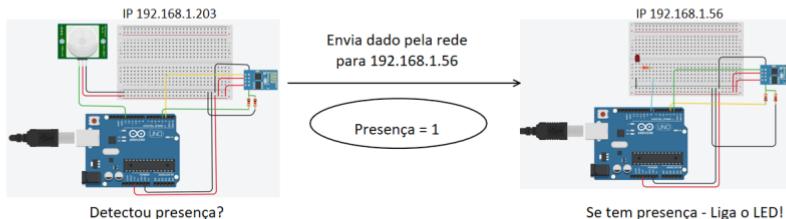


Veja como realizar a interface do Arduino UNO com uma Ponte H e o módulo ESP8266 neste QR code.

Comunicação Máquina-a-Máquina

A comunicação máquina-a-máquina, também abreviada por M2M, é quando dispositivos interagem automaticamente trocando mensagens, dados e/ou comandos entre si. Para isso, é necessário que ambos os elementos sejam capazes de entender as mensagens que serão enviadas e/ou recebidas entre as partes. Além disso, deve ser identificado ou referenciado pela rede. Em se tratando de dispositivos conectados tanto pela Internet como por uma rede local, a principal forma de referenciar e identificar quem é por meio do endereço IP de cada dispositivo. Uma vez que são conhecidos os endereços IP associados às partes envolvidas, um dispositivo pode enviar um dado comando ou valor de sensor a um outro dispositivo, sabendo o endereço IP deste. A Figura 2.18 ilustra um exemplo de interação Máquina-a-Máquina.

Figura 2.18 | Exemplo de interação Máquina-a-Máquina onde um sensor de presença interage com um LED



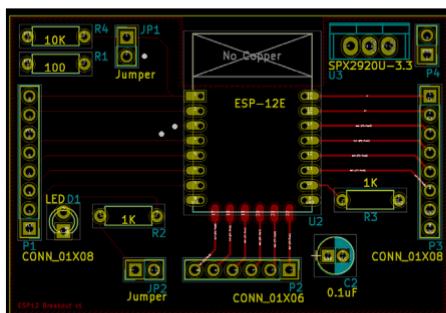
Fonte: elaborada pelo autor.

O ESP8266 possui suporte a comandos que especificam o endereço de rede que deverá receber determinado dado a ser enviado.

Construindo na prática sensores e atuadores

Montagens como as que foram feitas no TinkerCad são protótipos e podem ser caracterizadas como “provas de conceito”, onde você basicamente monta, conecta e programa os componentes em uso e demonstra a montagem em funcionamento. Uma vez que o setup foi validado, é hora de confeccionar placas de circuito que vão incorporar os componentes utilizados para determinado fim. Essas placas de circuito, normalmente abreviadas por PCB, possuirão tão somente os componentes necessários para operação do sistema. Para desenvolver essas placas de circuito impresso, é preciso fazer uso de programas do tipo CAD, que permitirão o desenvolvimento do *layout* das placas. E uma vez feito o *layout*, é preciso enviá-lo para um fabricante produzir a placa. Veja na Figura 2.19 um exemplo de *layout* de placa de circuito impresso.

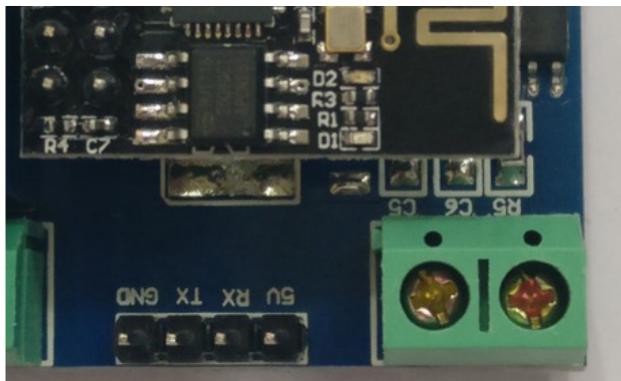
Figura 2.19 | Exemplo de placa para ESP8266 feita em aplicação do tipo CAD



Fonte: <https://i.stack.imgur.com/qJ93G.png>. Acesso em: 26 set. 2019.

A Figura 2.20 ilustra o exemplo de uma placa com relé juntamente com um ESP8266. O relé é o componente de cor azul na imagem, e serve para ligar e desligar equipamentos elétricos tal como um interruptor de parede. É um perfeito exemplo de um **atuador** conectado, pois o relé será remotamente controlado pelo ESP8266 presente na placa.

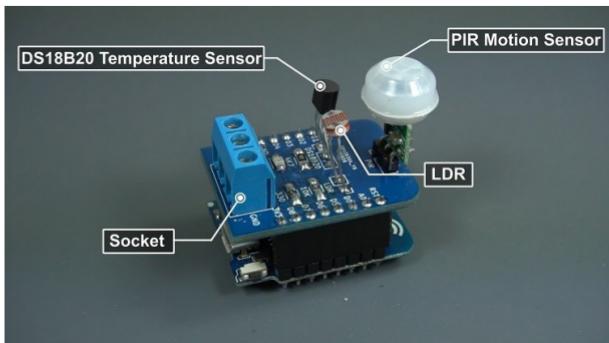
Figura 2.20 | Placa com relé para acionamento de cargas com ESP8266



Fonte: <http://www.alselectro.com/esp8266--01-iot-relay-board.html>. Acesso em: 26 set. 2019.

A Figura 2.21 mostra uma placa de sensores integrados a um ESP8266, fornecendo-nos um exemplo de **sensor** conectado.

Figura 2.21 | Placa com sensores de temperatura, presença e iluminação conectados a ESP8266 (localizado embaixo)



Fonte: <https://randomnerdtutorials.com/esp8266-multisensor-shield-with-node-red/#more-73094>. Acesso em: 26 set. 2019.

Observe que não há uma protoboard. Não há fios suspensos ou objetos soltos. Os elementos estão soldados, fixamente presos na placa, e tentando usar da melhor

forma possível o espaço da placa de circuito impresso. O mundo de sistemas embarcados e conectividade é um mundo em constante expansão. Mas ao passar pelo panorama apresentado foi possível mostrar os princípios básicos a respeito de conectividade, leitura de sinais digitais e analógicos de sensores, controle de atuadores de forma direta ou com drivers, além de observar como é a base da interação entre sistemas conectados que enviam e recebem dados em rede. Por fim, mostramos como são feitos na prática sensores e atuadores conectados.

Sem medo de errar

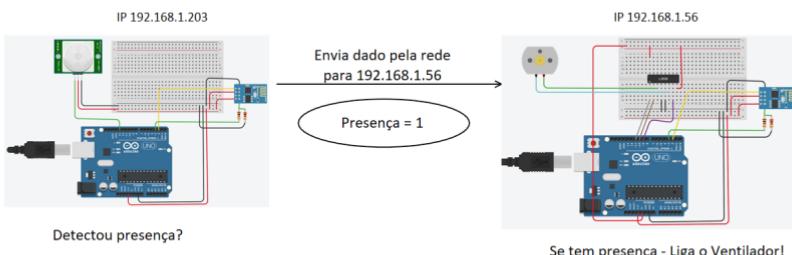
Fazendo uso de sensores e atuadores conectados, devemos arquitetar um sistema capaz de monitorar se há ou não presença de pessoas em uma dada sala e, de acordo com o status de presença reportado pelo sensor, devemos ligar ou desligar o ventilador presente nessa mesma sala.

Como o elemento sensor precisa reportar a presença da sala, utilizaremos um sensor de presença do tipo PIR, que é capaz de detectar se há pessoas ou não. O Arduino será programado de forma a ler o estado do sensor e enviar na rede o estado lido, por meio do ESP8266 ao atuador, este conectado na mesma rede.

Já o elemento atuador precisa controlar o ventilador presente na sala. Considerando que seja um ventilador elétrico de corrente contínua, pode-se adotar um setup, onde temos um driver de potência, uma Ponte-H, capaz de controlar a movimentação do motor elétrico do ventilador. Sem esquecer, claro, do Arduino que receberá o comando remoto por meio do ESP8266, controlando assim a Ponte-H para ligar ou desligar o ventilador.

Observe na Figura 2.22 como será o arranjo e um esquema de operação entre o sensor e o atuador conectados, interagindo entre si de forma que quando uma presença for detectada na sala, o sensor irá reportar ao atuador pela rede, e este por sua vez irá acionar o ventilador.

Figura 2.22 | Exemplo do sensor e atuador conectados para detectar presença e acionar o ventilador da sala



Fonte: elaborada pelo autor.

Em termos de código de programa que tanto sensor quanto atuador deverão executar, pode-se considerar que são basicamente os mesmos dos apresentados em cada exemplo ao longo do texto.

Avançando na prática

Controle dinâmico de um ventilador com base na temperatura e presença de pessoas na sala

Somente o liga e desliga do ventilador não é algo muito cômodo, pois já pensou se o ventilador for ligado em sua velocidade máxima em um dia mais frio, por exemplo? É muito mais interessante que o sistema seja “inteligente” ao ponto de que quando ligar o ventilador, sua velocidade seja proporcional à temperatura ambiente, de forma que quanto mais quente estiver, maior será a velocidade, e vice-versa, ou seja, quanto mais frio estiver, menor será a velocidade do ventilador.

Como integrar a leitura da temperatura no controle do ventilador, que já conta com a leitura de presença?

Resolução da situação-problema

A solução para este problema está em saber como incorporar o sensor de temperatura no sistema, fazendo com que envie os dados para o atuador responsável pelo controle do ventilador.

No atuador responsável pelo controle do ventilador haverá uma mudança de lógica para controlar o sinal do motor elétrico pela Ponte-H de forma proporcional, não mais simplesmente por meio de sinais ALTO e BAIXO, simplesmente, mas sim por meio de um sinal modulável chamado PWM.

Pinos no Arduino que possuem o sinal “~” são pinos que podem executar esse tipo de sinal modulável, que por sua vez é ativado no programa por meio da função **analogWrite()**, informando o pino e a intensidade desejada, que pode ser de 0 a 255.

No trecho de código em Linguagem C apresentado a seguir é mostrado um esqueleto do laço de execução do programa do atuador responsável pelo ventilador.

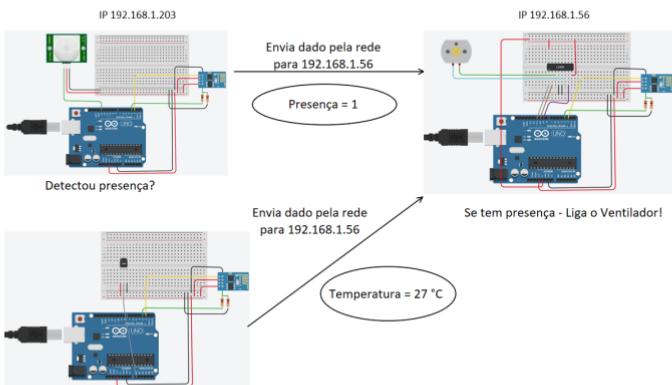
```
void loop() {  
    // Recebe dados da rede pelo esp8266  
    // filtrando os valores recebidos de uma
```

```
// e outro sensor  
int presenca = recebeDadosEsp8266_presenca();  
int temp = recebeDadosEsp8266_temperatura();  
// Faz escala da temperatura lida entre 0 a 50 graus Celsius  
// e controle de intensidade entre 0 e 255  
int val = map(temp, 0, 50, 0, 255);  
// Verifica se dado eh 1 ou 0  
// para ligar ou desligar o motor  
if (presenca == 1) {  
    analogWrite(11, val); // liga motor com intensidade "val"  
} else if (dado == 0) {  
    digitalWrite(11, LOW); // desliga motor  
}  
// aguarda 1 segundo ate verificar novamente  
delay(1000);  
}
```

Como temos dados vindos de 2 sensores diferentes, será preciso adotar rotinas que filtrem os dados recebidos pelo ESP8266, como destacado de forma abstrata no código apresentado.

A Figura 2.23 destaca como deve ser a interação entre os sensores de presença e de temperatura com o atuador.

Figura 2.23 | Arranjo de sensores de presença e temperatura e atuador de controle de motor elétrico conectados



Fonte: elaborada pelo autor.

Faça valer a pena

1. Você foi encarregado de realizar uma montagem com uma placa Arduino que trabalha com saídas digitais que operam a 5 V. Nessa montagem, é necessário conectar um LED vermelho que opera com tensão de 2.2 V e consome 20 mA.

No contexto apresentado, é correto assinalar o seguinte valor de resistência para o LED a ser conectado na montagem com Arduino:

- a. 1 K Ohm.
- b. 4.7 K Ohm.
- c. 120 Ohm.
- d. 140 Ohm.
- e. 220 Ohm.

2. Um sensor analógico do tipo TMP36 opera fornecendo uma tensão elétrica proporcional a 10 mV para cada °C medido, dentro da sua janela de operação que vai de -50°C a +125°C. O mesmo possui um ponto-fixo de partida (offset) de 0.5 V, fazendo com que tais medidas correspondam a um intervalo de 0 a 1.75 V. Sabendo disso, caso seja medida na saída do sensor TMP36 uma tensão de 0.78 V, essa tensão corresponde a qual temperatura lida pelo sensor?

No contexto apresentado, é correto assinalar o seguinte valor de temperatura lida pelo sensor analógico TMP36:

- a. 780 °C.
- b. 78 °C.
- c. 28 °C.
- d. 18 °C.
- e. 58 °C.

3. Um conversor analógico-digital é um elemento que converte um sinal elétrico analógico em uma representação digital para uso em sistemas embarcados. Uma das propriedades mais importantes de um conversor analógico-digital é a sua resolução, que indica o grau de escala que o dispositivo consegue equiparar um sinal analógico a um correspondente valor numérico digital. A título de exemplo, o Arduino possui um conversor analógico-digital com resolução de 10 bits, trabalhando num intervalo de tensão elétrica de 0 a 5 V. Isso significa que o conversor analógico-digital do Arduino é capaz de pegar um sinal elétrico analógico dentro de uma janela de 0 a 5 V e convertê-lo em um valor numérico de 0 a 1023 (10 bits). Dessa forma, caso seja lido um sinal de 0 V, seu correspondente digital será 0. E caso seja lido um sinal de 5 V, seu correspondente digital será 1023. Proporcionalmente, um sinal de 2.5 V resultaria em um valor numérico de 512. Assim, sabendo a equiparação entre uma tensão elétrica analógica e seu correspondente numérico digital, sistemas embarcados conseguem interagir com sensores e componentes analógicos.

No contexto apresentado, é correto assinalar que o conversor analógico-digital do Arduino tem a seguinte proporção de conversão entre tensão elétrica e unidade digital:

- a. 10 mV / unidade.
- b. 4.88 mV / unidade.
- c. 5.12 mV / unidade.
- d. 0.488 mV / unidade.
- e. 0.512 mV / unidade.

Seção 3

Programação em Python como ferramenta para IoT

Diálogo aberto

Prezado estudante,

Chegamos então à seção final da unidade após ter trabalhado conceitos de microcontroladores e sensores fazendo uso da formidável ferramenta que é o Arduino. Agora é hora de avançar um pouco mais no mundo da Internet das Coisas, e nada melhor do que fazer uso da Linguagem Python como instrumento deste avanço.

Python é uma das Linguagens mais utilizadas no meio da Internet das Coisas porque facilmente permite a **integração e manipulação** de diferentes fontes e formatos de dados. E não só isso: Por ser uma Linguagem de Alto Nível, abstrai muitas das complexidades por trás de rotinas de manipulação de dados e recursos de rede. Em poucas linhas de código você será capaz de escrever uma aplicação que realiza leitura de dados de sensores e o envio desses dados para nuvem, aos moldes da Internet das Coisas.

Todavia, há um porém... Python é uma Linguagem que demanda mais recursos computacionais para ser executada, de forma que sistemas embarcados mais “simples”, como o Arduino UNO, por exemplo, não são capazes de executar código Python. Mas fique tranquilo! Existem sistemas, meios e abordagens de utilizar a Linguagem – seja usando um computador que “conversa” com o Arduino, seja utilizando um sistema embarcado mais sofisticado, como uma Raspberry Pi (OLIVEIRA, 2017).

De forma a visualizar o poder do Python na prática, vamos considerar um caso em que se tenham sensores conectados para leitura de sinais de temperatura e umidade de um dado ambiente, e atuadores também conectados ligados a ventiladores e umidificadores.

O funcionamento do sistema deve ser tal que se os sensores de temperatura apontarem uma temperatura acima de um dado valor de referência, o atuador do ventilador deve ser ligado!

E caso a temperatura esteja abaixo do valor de referência, o ventilador deve permanecer desligado. Analogamente o funcionamento é semelhante para o controle de umidade com o umidificador.

Então, como realizar essa interface em Python? Como fazer a interface com sensores e atuadores? Como conectar com serviços de nuvem?

Primeiramente, é necessário compreender como é e como opera um código Python, valendo-se de como a linguagem é estruturada desde variáveis, condicionais, laços de repetição, métodos e objetos (lembrando que será apenas uma introdução básica, caso queria se aprofundar em Python, use a bibliografia recomendada). E, na sequência, veremos como criar elementos “virtuais” para simular nossos sensores e atuadores, além de utilizar módulos adicionais para permitir a integração do código Python com a nuvem aos moldes da Internet das Coisas.

Não pode faltar

A Internet das Coisas é uma onda que veio para ficar, e se consolida ainda mais em uma sociedade já conectada que se beneficia com a presença de **coisas** também conectadas (COELHO, 2017). Todavia, o Mercado demanda uma velocidade cada vez mais maior e desafiadora para entrega de novos produtos e soluções. Dessa forma, torna-se necessário fazer uso de linguagens de programação que permitam a experimentação ou o desenvolvimento de soluções capazes de atender os requisitos necessário de forma mais prática e ágil. E a Linguagem Python atende a essas expectativas.

Python é uma Linguagem Interpretada, ou seja, seu código não é compilado, mas executado linha a linha sequencialmente por meio de um interpretador. O ponto mais positivo disso é que códigos em Linguagem Python são multiplataforma, pois rodam executados por interpretadores Python. E o ponto negativo disso é a *velocidade* – como o código é interpretado, por baixo dos panos dos sistemas computacionais, a velocidade de execução é menor do que a de códigos compilados (PERKOVIC, 2016).

Python também é uma Linguagem aberta e gratuita, ou seja, não é necessário pagar nenhuma licença pelo seu uso. Sua abertura, somada ao seu alto nível e facilidade de uso, fez com que a linguagem fosse amplamente adotada, possuindo ampla gama de usuários, desenvolvedores e bibliotecas de suporte para os mais diversos fins (PERKOVIC, 2016). Seja você usuário de sistemas Windows, Linux, Mac e até mesmo Android, saiba que o Python é compatível com as principais plataformas em uso atualmente. Sua instalação é fácil, basta ir ao site do Projeto (PYTHON, 2019), baixar o programa instalador para sua plataforma e proceder com a instalação em seu computador pessoal e pronto!

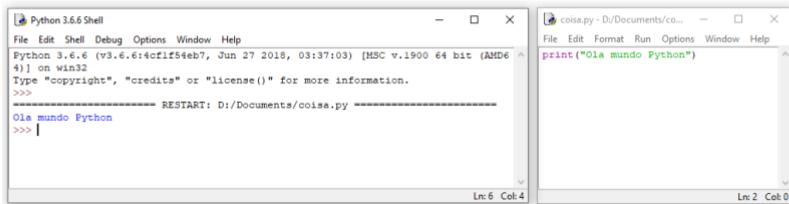
São duas as principais ferramentas quando se está desenvolvendo código Python:

Interpretador de Código – Representado por um programa na forma de *Shell*, ou console, no qual os comandos e códigos podem ser executados controladamente e seletivamente.

Editor de código – Onde é possível abrir, criar ou editar códigos a serem executados pelo Shell Interpretador.

Veja na Figura 2.24 um exemplo de “Ola mundo” em Python, onde tanto o Shell Interpretador e o Editor aparecem lado a lado.

Figura 2.24 | Exemplo do Python Shell, à esquerda, com o editor Python, à direita



```
Python 3.6.6 (v3.6.6:cfef54eb7, Jun 27 2018, 03:17:03) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> print("Ola mundo Python")
>>> |
```

```
File Edit Format Run Options Window Help
coisa.py - D:/Documents/co...
print("Ola mundo Python")
```

Fonte: elaborada pelo autor.

Observe que não foi necessário “importar” nenhuma biblioteca adicional, pois Python já possui embutido nativamente suporte aos recursos para exibir mensagens na tela, e até mesmo para realizar leitura de comandos do usuário, dentre outros.

Assimile

Python, semelhantemente a outras linguagens de programação, também possui muitas funções de suporte nativamente embutidas, tais como `print()` para exibir mensagens no Console Shell, como mostrado na Figura 2.24, e até mesmo `input()` para ler dados do usuário.

Uma das coisas que mais “assusta” os desenvolvedores iniciantes de Python, que por sua vez já tiveram uma dada familiaridade com outras Linguagens de Programação tal como C ou Java, é que Python é uma Linguagem caracterizada pela **inferência de tipos**.

Outro ponto muito interessante da Linguagem Python é que não há chaves na delimitação dos métodos e laços de repetição. A estrutura de código é organizada por meio da **indentação**.

Assimile

Indentação é aquele espaçamento entre o início da margem e o texto, seja num documento de texto, propriamente, seja num documento de código-fonte (PRIBERAM, 2019). No caso da Linguagem Python, sequências de código são delimitadas pelo fato de estarem organizadas em um mesmo espaçamento. Adiante, veja como é tratada a questão da indentaçāo em Python.

codigo

outro codigo

outro outro codigo

Assim como Java e C, Python também possui estrutura de laços condicionais. Essa estrutura permite criar fluxos de decisão no programa, de forma que se determinada variável assumir um ou outro valor, certas ações serão tomadas para cada caso programaticamente. Citando o exemplo dos sensores e atuadores, suponhamos que o atuador tenha que fazer uma ação **se** a temperatura estiver maior que um determinado valor, e outra ação **se** a temperatura estiver abaixo de um determinado valor. Estes são casos de ações condicionais.

Em se tratando de Python, os laços condicionais usam a palavra reservada **if**, seguida da condição que se quer avaliar. Lembre-se que cada sentença condicional em Python acaba com “**:**”. Caso outra condição seja avaliada dentro de um mesmo laço, é usada a palavra reservada **elif**. Por fim, caso alguma condição padrão seja avaliada caso nenhuma das demais tenha sido atendida, é usada a palavra reservada **else** (PERKOVIC, 2016). Veja como fica uma sequência aninhada para avaliar os possíveis valores de uma determinada variável a:

a = 42

Verifica se valor da variavel eh menor que 42 e exibe mensagem

if a < 42:

print(“a eh menor que 42”)

Verifica se valor da variavel “a” eh maior que 42 e exibe mensagem

elif a > 42:

print(“a eh maior a 42”)

Caso onde nenhuma das alternativas anteriores seja atendida, e exibe mensagem

else:

```
print("a eh igual a 42")
```

Os laços condicionais ajudam na execução de rotinas com base em valores, ou melhor, **condições** diversas, o que é muito útil em se tratando de Internet das Coisas – pois muitas das decisões são tomadas com base em valores de referência frente aos dados lidos de sensores ou comandos recebidos da nuvem.

E para executar sequencialmente uma dada operação, seja por um intervalo de valores determinado, ou indeterminadamente, existem os chamados **laços de repetição**. Para a Linguagem Python temos tanto **for** como **while** como palavras reservadas para comandos de laços de repetição, cada qual com as suas peculiaridades, a saber:

- **for** – interessante para trabalhar rotinas de código sobre um vetor de dados, um intervalo determinado e conhecido de valores. Útil na aplicação de filtros ou demais operações matemáticas sobre um vetor carregado de dados.
- **while** – interessante para trabalhar uma determinada execução de código enquanto uma dada condição for verdadeira. Útil quando deseja-se executar repetidamente uma porção do código até que determinada condição deixe de ser verdade.

Exemplificando

No caso da Linguagem Python, o laço **for** trabalha um intervalo de valores que irá percorrer um a um dos valores dentro do intervalo (PERKOVIC, 2016). Para gerar automaticamente um intervalo de valores, é possível utilizar a função do Python **range**, como mostrado adiante:

```
# Realizar o laço “for” com base nos valores de i no intervalo de 0 a 10.  
for i in range(0,10):  
    # Exibe a mensagem seguida do valor da variavel i.  
    print("IoT com Python eh Legal - ", i)
```

Observe que comentários em Python começam com o caractere “#”. Evite comentários em Python com caracteres especiais, como “á”, “ç”.

Vale considerar que é de suma importância tomar cuidado com os valores adotados como referência para início e/ou parada dos laços condicionais.

Normalmente, tais valores são obtidos de forma empírica quando se está testando a interação com algum elemento **real**, no âmbito de sistemas embarcados / Internet das Coisas, ou de forma previamente arquitetada com base nos requisitos da aplicação.

Exemplificando

Para usar o laço **while**, diferentemente do **for**, é necessário que a variável de referência a ser avaliada na condição do laço tenha sido previamente inicializada! Veja adiante um exemplo de uso do **while**, com efeito semelhante ao que foi feito com o uso de **for** anteriormente.

```
# Inicializa a variavel i com valor 0.  
i = 0  
# Realiza laço while enquanto valor de i for menor que 10.  
while (i < 10):  
    # Exibe a mensagem seguida do valor da variavel i.  
    print("IoT com Python eh Legal - ", i)  
    # Soma 1 ao valor da variavel i.  
    i = i + 1
```

Outro recurso presente em toda sorte de linguagens de programação, e também em Python, são as funções ou métodos. Elas são de suma importância porque auxiliam no reuso de código, ao passo que rotinas comumente utilizadas ou chamadas podem ser agrupadas em funções, que por sua vez podem ser adequadamente chamadas em outros trechos do código de programa.

Funções ou métodos em Python sempre começam com a palavra reservada **def**, seguido do nome da função e seus parâmetros entre parênteses, caso tenha. Por fim, a função pode executar e/ou retornar algum dado ou operação por meio do comando **return**. Veja adiante o exemplo de declaração da função **soma**, seguido da sua chamada em código.

```
# Definicao da funcao soma com parametros a e b, que retorna a soma  
dos parametros  
def soma(a, b):  
    return a + b  
  
# Chama a funcao soma com parametros 10 e 5, e o resultado eh  
armazenado na  
# variavel resultado.
```

```
resultado = soma(10, 5)
# Exibe o texto no console com o valor da variavel resultado.
print("O resultado da soma eh: %d" % resultado)
```

Outra vantagem do Python é a grande quantidade de bibliotecas e módulos disponível para uso, sejam nativamente disponíveis como parte da própria Linguagem, sejam bibliotecas de terceiros. Uma das bibliotecas que é muito utilizada em toda sorte de códigos Python é a biblioteca **time**, que permite manipulações de tempo em código Python. Para incorporar módulos (ou bibliotecas) em códigos Python, fazemos uso da palavra reservada **import**, seguida do nome da biblioteca desejada.

No caso de criar um laço de repetição **infinito**, ou seja, que ficará em execução indefinidamente, usa-se o termo **booleano True**. Para “segurar” a execução vez após vez do laço de repetição por um período de tempo, como **1 segundo**, usa-se a biblioteca **time** e sua função **sleep**, como mostrado no trecho de código adiante.

```
# Importa a variavel time ao programa Python
```

```
import time
```

```
# Ira executar o código indefinidamente, “enquanto verdade”
```

```
while True:
```

```
    # Ira segurar a execucao do codigo por (1) segundo
```

```
    time.sleep(1)
```

```
    # Exibe a mensagem no console
```

```
    print("Executando...")
```

Quando se trata de Internet das Coisas, o foco é controlar as... Coisas! E como fazer isso via Python? Bom, temos duas formas:

1. Realizar uma interface no programa Python sedo executado em um computador com algum dispositivo físico que interaja com sensores e atuadores, como o Arduino.
2. Executar o programa Python diretamente em um sistema que tenha suporte à interface com sensores e atuadores, como a Raspberry Pi.

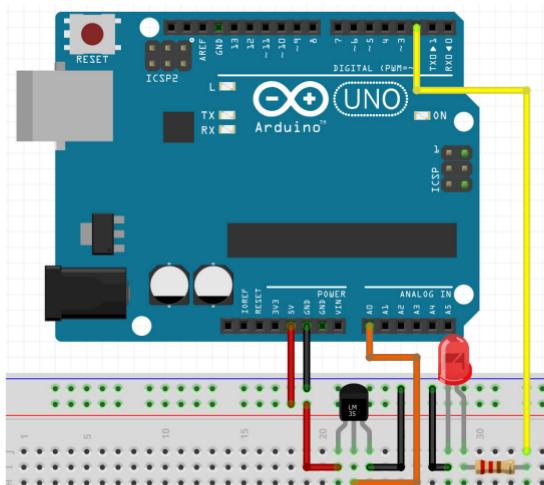
Na presente seção e unidade trabalharemos com a primeira forma, dada a já familiaridade do leitor com o Arduino. E para realizar essa interface

entre um programa Python rodando em um computador conectado com um Arduino, fazemos uso da biblioteca Python **pySerial** (PYSERIAL, 2019). Essa biblioteca facilita o envio e recepção de mensagens entre o computador e o Arduino por meio de uma porta Serial de comunicação.

Quando se trata dessa forma de interface, é necessário cuidar, então, tanto da programação que opera no dispositivo físico, no nosso caso um Arduino, como do programa Python a ser executado no computador que **conversará** com o Arduino.

Assim, pensando no cenário da Situação-Problema, imagine um Arduino realizando interface com um sensor analógico de temperatura LM35 na Porta Analógica A0, e controlando um LED por meio da Porta Digital 2, semelhante ao arranjo mostrado na Figura 2.25.

Figura 2.25 | Arduino realizando interface com Sensor LM35 e LED



Fonte: elaborada pelo autor.

Para que o Arduino do arranjo mostrado na Figura 2.25 seja capaz de ler os sinais do sensor de temperatura e controlar o LED conectados a ele, e também receber comandos e enviar os dados de sensor a um computador conectado via Serial, podemos programar o Arduino com o seguinte código, observando os comentários de suas partes:

// Definicao das constantes e variaveis a serem utilizadas

```
const int LED = 2;
```

```
const int LM35 = A0;
```

```
float temperatura;
char comando;

// Setup - executada assim que o Arduino eh inicializado e configura
elementos do microcontrolador

void setup() {
    Serial.begin(115200); // Inicializa a comunicação serial
    pinMode(LED, OUTPUT); // Configura sinal do LED como saída.
}

// loop - Rotina a ser executada indefinidamente pelo Arduino
void loop() {
    if (Serial.available() > 0) {
        // Realiza leitura de letra que chegou na serial
        comando = (char) Serial.read();
        if (comando == 'L') {
            digitalWrite(LED, HIGH); // Se a letra eh L, liga o LED
        }
        else if (comando == 'D') {
            digitalWrite(LED, LOW); // Se a letra eh D, Desliga o LED
        }
    }

    // Converte a leitura do sensor na escala de 10mV/grau Celsius
    // e considerando a conversao analogico-digital de 10bit do Arduino
    temperatura = (float(analogRead(LM35))*5/(1023))/0.01;
    Serial.println(temperatura);
    // Segura a execucao do programa por 5 segundos
```

```
delay(5000);
```

```
}
```

Do lado do Computador, executando um código Python, podemos colocar o seguinte código para ser executado de forma a interagir com os dados de sensor enviados pelo Arduino, e comandar o Arduino para ligar ou desligar um determinado atuador (nesse caso, um LED, mas poderia ser um ventilador, aquecedor etc.).

```
# Importa a biblioteca serial (da pyserial)
```

```
# e a biblioteca time, necessarias para o programa
```

```
import serial
```

```
import time
```

```
# Cria um elemento de comunicacao serial com o Arduino, na porta COM3  
e velocidade de 115200 bps.
```

```
arduino = serial.Serial("COM3", 115200)
```

```
# Laco infinito de execucao do programa
```

```
while True:
```

```
    # Le resposta via serial do Arduino
```

```
    resposta = arduino.readline()
```

```
    # Converte valor da resposta em inteiro para podermos comparar
```

```
    resposta_int = int(resposta)
```

```
    # Verifica os valores dos sensores
```

```
    if (resposta_int > 40):
```

```
        arduino.write('L') # Manda comando L pro Arduino - Ligar
```

```
    else:
```

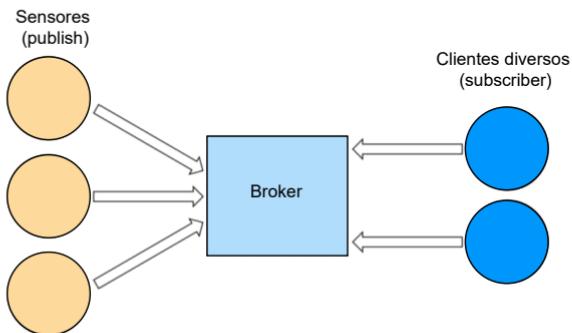
```
        arduino.write('D') # Manda comando D pro Arduino - Desligar
```

```
    # Aguarda 5 segundos para proxima execucao
```

```
    time.sleep(5)
```

E em se tratando de conectividade com Internet das Coisas, a principal biblioteca para IoT com Python é a **paho-mqtt** (ECLIPSE, 2019), que é voltada para interação entre elementos utilizando o protocolo MQTT (MQTT, 2019). Como mostrado na Figura 2.26, MQTT faz uso de um Broker, que é um elemento que atua como intermediário entre o fluxo de mensagens geradas por produtores de dados (como por exemplo, nós sensores) e consumidores de dados (nós clientes, ou “assinantes”, como por exemplo, atuadores). A biblioteca **paho-mqtt** implementa os recursos necessários para publicar mensagens ou responder a mensagens publicadas. O fluxo de mensagens em MQTT se dá por meio de tópicos, que denotam o “caminho” do dado em questão, como **kroton/iot/laboratorio/temperatura**.

Figura 2.26 | Estrutura de operação do MQTT com Broker, clientes para publicação (publish) e subscriber (inscrição)



Fonte: <https://www.embarcados.com.br/mqtt-protocolos-para-iot/>. Acesso em: 12 fev. 2019.

Pesquise mais

Criado pela IBM, o Message Queue Telemetry Transport (MQTT) é um dos protocolos de telemetria mais utilizados atualmente.

BARROS, M. MQTT – Protocolos para IoT. Embarcados, 26 mar. 2018.

Veja s seguir um modelo de aplicação Python para IoT com a biblioteca **paho-mqtt**.

A estrutura-base de um código-fonte Python com **paho-mqtt** é semelhante à apresentada adiante. Observe que é necessário ter métodos **on_connect** e **on_message** para lidar com eventos de conexão e de mensagem. E para envio de mensagens ao Broker em um dado tópico, usa-se o método **publish** da biblioteca **paho-mqtt**.

```

# Importa a biblioteca paho mqtt e time para uso na aplicacao
import paho.mqtt.client as mqtt
import time

# Definicao de elementos necessarios no programa, como url do broker mqtt
# porta de conexao com broker, tempo de verificacao da conexao (keepalive)
# e caminho do topico mqtt a ser utilizado
Broker = "iot.eclipse.org"
Porta = 1883
KeepAlive = 60
Topico_Temp = "kroton/iot/laboratorio/temperatura"

# Callback de conexao ao broker – Esse metodo eh chamado pelo paho mqtt
# quando eh realizada uma conexao. Serve para indicar se a conexao foi
realizada
# com sucesso ou nao.
def on_connect(client, userdata, flags, rc):
    print("Conectado. Status: "+str(rc))
    client.subscribe(Topico_Temp)

# Callback de mensagem recebida do broker. Esse metodo eh utilizado pelo
paho mqtt
# toda vez que uma mensagem chegou a um topico que foi subscrito pelo
cliente, de
# forma que o cliente saiba que chegou uma mensagem, e assim seja capaz
de tomar
# alguma acao.
# O atributo msg eh que fornece o topico e a mensagem (payload) transpor-
tada.
def on_message(client, userdata, msg):
    msg_recv = str(msg.payload, 'utf-8')
    print("Topico: " + msg.topic + " | Msg: " + msg_recv)

# Sequencia principal do codigo da aplicacao.
print("Inicializando MQTT...")
# Inicializa o componente de cliente MQTT
client = mqtt.Client()
# Vincula os metodos de conexao e de mensagem recebida
client.on_connect = on_connect
client.on_message = on_message
# Realiza a conexao com o broker, informando os parametros utilizados
client.connect(Broker, Porta, KeepAlive)

```

```
# Inicializa o monitor de mensagens com o broker
client.loop_start()
while True:
    # Publica a mensagem 42 no caminho informado pela variavel Topico_Temp
    client.publish(Topico_Temp, "42")
    # Segura a execucao do programa por 5 segundos
    time.sleep(5)
```

MQTT é um protocolo formidável para IoT, todavia existem outros tais como HTTP e AMQP, também contando com suporte para Linguagem Python, a saber, temos as bibliotecas `httplib` e `urllib`, nativas do Python para HTTP, e `Pika` para AMQP (PIKA, 2019). Porém, MQTT é o protocolo mais comumente adotado atualmente para IoT, assim como a biblioteca `paho-mqtt` é a mais utilizada para MQTT em Python.

Pesquise mais

Protocolos para IoT – MQTT vs HTTP. Qual é o melhor?

- YUAN, Michael. **Conhecendo o MQTT**. IBM, 4 out. 2017.
- LORENZO, Jose Luiz Sanchez. **DEAL. IoT – MQTT e REST, os protocolos utilizados no mundo IoT**. Deal, 7 maio 2019.

Em se tratando de intercâmbio de dados entre elementos sensores e atuadores, dentre outros, é importante também pensar e adotar um formato comum de apresentação de dados. Atualmente, JSON tem-se consagrado como padrão por apresentar uma estrutura de dados mais enxuta, adequada a um cenário em que milhares de milhares de dispositivos transmitem dados ao mesmo tempo, e qualquer byte a menos pode melhorar o uso da banda de rede.

Reflita

Em se tratando de Internet das Coisas, menos é mais. Ou seja, quanto mais enxuto e direto for uma estrutura de dados e um protocolo, melhor. Dessa forma, MQTT é realmente um protocolo melhor para transmissão de dados do que HTTP, já consagrado na Web? E para representação de dados, JSON é melhor do que XML?

Python é uma elegante e poderosa Linguagem de Programação. Porém, ainda não é a Linguagem mais voltada para programação de sistemas embarcados de mais baixo nível, que operam com poucos MHz e MegaBytes de

RAM e Flash, como o caso do Arduino, utilizado como elemento sensor e/ou atuador no mundo da Internet das Coisas. Dessa forma, como visto no caso de interagir com dispositivos como Arduino, é necessário realizar a **interface** com sistemas mais sofisticados, como um computador, enquanto que o Arduino em si continua executando código em C. Isso ocorre porque Python é uma linguagem interpretada, orientada a objetos, e que faz uso de recursos que demandam um poderio computacional maior.

Isso serve de lição para aprender mais e mais sobre programação e sobre qual a melhor linguagem a ser adotada em cada sistema – Sistema Embarcado em C, Computador com Python, e também sobre formas de fazer sistemas diferentes conversarem entre si – seja serial, seja pela internet usando a nuvem e MQTT.

Na próxima unidade, você trabalhará com Raspberry Pi, que é um Sistema Embarcado mais sofisticado, capaz de executar Python nativamente. Quando tiver a oportunidade, compare a forma de operação e execução entre o Arduino e a Raspberry Pi, e você verá como são diferentes!

Bons estudos!

Sem medo de errar

Fazendo uso de Python, devemos arquitetar uma solução envolvendo elementos sensores e elementos atuadores que vão se comunicar entre si por meio de serviços da Nuvem aos moldes da Internet das Coisas, de forma que os elementos atuadores tomem determinadas ações com base nos dados enviados dos sensores pela nuvem.

Tomando por base o protocolo MQTT como meio de troca de dados entre sensores e atuadores, conectados em um Broker, deve-se ter em mente que:

- Elementos sensores publicarão em um tópico relacionado ao domínio da informação relacionada aos dados coletados.
- Elementos atuadores subscreverão aos tópicos que os elementos sensores estarão publicando dados → Dessa forma, os atuadores serão notificados pelo **broker** quando os sensores publicarem novos dados.

A estrutura-base, tanto da aplicação Python dos sensores como dos atuadores, será bem parecida ao modelo de aplicação com **paho-mqtt** mostrado anteriormente, salvo modificações de um lado para focar em envio de dados (sensores), e de outro na resposta aos dados recebidos (atuadores).

De forma geral, uma aplicação **sensor simulada** ou **virtual** deverá gerar aleatoriamente dados de sensor em um intervalo de valores, coerente com

um sensor **real**. Pense em algo como, por exemplo, 0 a 100 graus Celsius para um sensor de temperatura. Para gerar esse perfil de valores aleatórios em Python, usa-se a biblioteca **random** da seguinte forma:

```
# Importa a biblioteca random, nativa do Python, necessaria para gerar elementos aleatorios.
```

```
import random
```

```
# Definicao da funcao leitura_sensor_temp, que retorna um valor inteiro aleatorio entre 0 e 100.
```

```
def leitura_sensor_temp():
```

```
    return random.randint(0,100)
```

```
# Exibe a mensagem no console com o valor retornado pela chamada da funcao "leitura_sensor_temp()"
```

```
print("Teste sensor: %d" % leitura_sensor_temp())
```

A cada chamada da função **leitura_sensor_temp**, um valor entre 0 e 100 será gerado.

Para usar essa função juntamente com a função **publish**, basta converter o retorno de tipo com **str()** do Python:

```
...
```

```
# Realiza uma acao de publish no caminho informado pela variavel Topico_Temp, convertendo em str() o valor
```

```
# retornado pela chamada de leitura_sensor_temp().
```

```
client.publish(Topico_Temp, str(leitura_sensor_temp()))
```

Já para o caso da aplicação do **atuador**, o segredo de sua funcionalidade está no tratamento das mensagens que chegam, com base no tópico que foi subscrito. Dessa forma, deve-se melhorar a rotina **on_message** para tratar os valores das mensagens que chegarem, como por exemplo:

```
def on_message(client, userdata, msg):  
    # Criacao da variavel msg_recv como sendo uma string (str) do conteudo  
    # da msg (payload) em  
    # formato utf-8.  
    msg_recv = str(msg.payload, 'utf-8')  
    # Exibicao de texto em console mostrando a mensagem que chegou e seu  
    # topico relacionado.  
    print("Topico: " + msg.topic + " | Msg: " + msg_recv)  
    # Para comparacao com if/else, convertemos o conteudo da mensagem  
    # msg_recv em inteiro por meio de int()  
    # do Python.  
    temperatura = int(msg_recv)  
    # Verificamos se o valor da "temperatura" eh maior que 40, para tomar  
    # uma dada acao, e outra acao caso  
    # contrario.  
    if (temperatura > 40):  
        print("Temperatura alta!")  
        # E aciona ventilador por meio de chamada de interface com atuador.  
    else:  
        print("Temperatura normal...")  
        # E desliga ventilador por meio de chamada de interface com atuador.
```

A função **int()** do Python é capaz de converter valores considerados **string** (ou texto) em formato Inteiro, necessário para as rotinas dos laços condicionais (**if/else**).

No caso de haver um relé conectado ao equipamento (no caso de uma Raspberry Pi), bastaria chamar os comandos Python para ligar ou desligar o sinal, conforme mostrado na forma de comentário para cada caso.

De forma análoga ao que foi feito e trabalhado para o **fictício** sensor de temperatura, também se aplica para o sensor de umidade e demais sensores que se fizerem necessários.

Filtrando ruídos de dados de sensores - Melhorando a resposta a eventos

Vamos supor que você esteja implementando um sistema real de sensores e atuadores conectados de forma que os sensores coletam dados de temperatura, umidade ou alguma outra grandeza física, e enviam os dados coletados a atuadores por meio de serviços na nuvem, fazendo uso do protocolo MQTT e Linguagem Python. Porém, algumas vezes os sensores podem apresentar medidas “erradas”, ruídos por assim dizer, ou simplesmente podem sofrer alguma falha de conexão e não enviar o dado esperado por algum tempo.

A título de exemplo, pense em um sensor que está enviando temperaturas em escala de graus Celsius tais como 10, 10, 11.5, 12.3, 11.8, 90, 12, 11.4. O valor 90, no meio da leitura, foi um pico “ruidoso”, e que pode ser mal interpretado. Se a temperatura estivesse tão elevada assim, os valores da sequência também estariam em escalas próximas. Já outro caso, também bem provável em se tratando de Coisas conectadas à Internet, é a de perda de conexão. O que fazer quando um sensor parar de enviar dados por um período de tempo?

Resolução da situação-problema

Vamos tratar cada caso, começando pelo caso do ruído. Sempre que for tratar sinais de sensores, adote filtros de sinais como, por exemplo, filtro da média, no qual é feito uma média de determinadas amostras de sinais coletados. É uma técnica simples, mas que atenua dados discrepantes com relação aos demais, evitando respostas inadequadas.

Tomando por base a Linguagem Python e um método para leitura de um dado sensor de temperatura; você pode fazer um filtro de média da seguinte forma:

```
filtro_media = 0
```

```
filtro_media = (filtro_media + leitura_sensor_temp()) / 2
```

E para o caso de o **sensor** não enviar dados por um período de tempo, o **atuador** deve **continuar** operando com base na **última leitura** enviada pelo sensor.

Isso remete ao fato de que a resposta aos eventos em **on_message** devem servir para alterar o estado de operação do **atuador**, cujo controle será executado por laços condicionais dentro do laço infinito (while True) da aplicação modelo MQTT com **paho-mqtt**, por exemplo, consideradas as modificações necessárias.

Faça valer a pena

1. Python tem sido uma linguagem cada vez mais adotada, seja na indústria, na academia e até mesmo por entusiastas, começando a dar sinais de que um dia poderá ocupar o trono outrora ocupado pela linguagem Java. Com amplas possibilidades de aplicação, Python tem sido adotado em sistemas web e até mesmo em sistemas embarcados com aplicações para Internet das Coisas.

Com base no que foi estudado sobre Python, avalie as afirmativas apresentadas a seguir.

- I. Python é uma Linguagem de Alto Nível, multiplataforma e capaz de ser executada em sistemas que vão desde um simples Arduino UNO a até supercomputadores.
- II. Python permite manipulação e integração de dados, o que tem favorecido sua adoção em sistemas voltados à Internet das Coisas.
- III. Python é uma linguagem interpretada, com desempenho superior a aplicações compiladas..

No contexto apresentado, é correto assinalar o que se afirma em:

- a. I, apenas.
- b. II, apenas.
- c. III, apenas.
- d. I e II, apenas.
- e. I, II e III.

2. Aplicações em código Python podem ser multiplataforma, ou seja, um mesmo código Python pode ser executado em computadores que estejam executando sistema-operacional Windows, Linux, Mac, dentre outros. Dessa forma, a maior restrição é que o Interpretador tenha suporte na plataforma desejada, considerando que o código será, por sua vez, interpretado.

Com base no que foi estudado sobre Python para IoT, avalie as afirmativas apresentadas a seguir.

- I. Um código Python pode realizar manipulações de baixo-nível, como controle de sinais e interface com sensores e atuadores, o que depende de arquitetura, como o caso de código aplicado à Raspberry Pi.
- II. Bibliotecas Python, tanto da Linguagem em si como de terceiros, tendem a ser mais genéricas, o que permite importar bibliotecas como **time** e **paho-mqtt** sem conflitos, seja em um ambiente Windows, seja Linux ou Mac.
- III. A Linguagem Python é fracamente tipada, ou seja, não exige a declaração de tipos de variáveis, como por exemplo **int x = 10**.

No contexto apresentado, é correto assinalar o que se afirma em:

- a. I e III, apenas.
- b. III, apenas.
- c. I e II, apenas.
- d. I, apenas.
- e. II e III, apenas.

3. MQTT é um protocolo criado pela IBM, baseado em TCP/IP, leve e enxuto, que tem sido cada vez mais adotado em aplicações para Internet das Coisas. Por ser aberto, há suporte para esse protocolo em diversas linguagens de programação, tal como Python, por meio da biblioteca paho-MQTT. Sobre esse protocolo, sua biblioteca e seu uso na linguagem Python, avalie as afirmações a seguir.

MQTT é um protocolo baseado em três elementos principais: elementos que publicam dados, elementos que assinam notificações sobre dados, e um Broker que atua como intermediário entre os elementos que publicam e subscrevem, e como canal de informação há a estrutura do tópico que opera semelhantemente a uma URL.

A biblioteca paho-mqtt fornece todo o aparato necessário para realizar conexão com o Broker, enviar mensagens e receber notificações em tópicos desejados por meio de métodos relacionados às ações de publish (publicar) e subscribe (subscrever, assinar).

Por ser baseado em TCP/IP, sempre haverá confirmação de entrega de pacotes, independente do QoS adotado em MQTT. No contexto apresentado, é correto assinalar o que se afirma em:

- a. I e II, apenas.
- b. II e III, apenas.
- c. I, apenas.
- d. III, apenas.
- e. I, II e III.

Referências

- AUTODESK. TINKERCAD. **Galeria de Circuitos**. Disponível em: <https://www.tinkercad.com/things?type=circuits>. Acesso em 26 set. 2019.
- BARROS, M. **MQTT – Protocolos para IoT**. Embarcados, 26 mar. 2018. Disponível em: <https://www.embarcados.com.br/mqtt-protocolos-para-iot/>. Acesso em: 9 jul. 2019.
- COELHO, P. **Internet das Coisas**: introdução prática. Lisboa: FCA, 2017.
- ECLIPSE. **Paho MAQTT**. Disponível em: <https://www.eclipse.org/paho/>. Acesso em: 9 jul. 2019.
- FLANAGAN, D. **JavaScript**: O Guia Definitivo. Porto Alegre: Bookman, 2014.
- INSTITUTO NEWTON C. BRAGA. **Como funcionam os Conversores A/D - parte 1 (ART224)**. Disponível em: <http://www.newtoncbraga.com.br/index.php/como-funciona/1508-conversores-ad>. Acesso em: 9 set. 2019.
- MQTT. Disponível em: <http://mqtt.org/>. Acesso em: 9 jul. 2019.
- OLIVEIRA, A. S. de; ANDRADE, F. S. de. **Sistemas Embarcados – Hardware e Firmware na Prática**. São Paulo: Érica, 2010.
- OLIVEIRA, C. L. V.; ZANETTI, H. A. P. **Arduino Descomplicado - Como Elaborar Projetos de Eletrônica**. São Paulo: Érica, 2015.
- OLIVEIRA, S. **Internet das Coisas com ESP8266, Arduino e Raspberry Pi**. São Paulo: Novatec, 2017.
- PERKOVIC, L. **Introdução à Computação Usando Python - um foco no desenvolvimento de aplicações**. São Paulo: LTC, 2016.
- PIKA. Pure Python RabbitMQ/AMQP 0-9-1 client library. Disponível em: <https://github.com/pika/pika>. Acesso em: 15 jul. 2019.
- PRIBERAM. **Indentação**. Disponível em: <https://dicionario.priberam.org/indenta%C3%A7%C3%A3o>. Acesso em: 9 jul. 2019.
- PYSERIAL. Disponível em: <https://github.com/pyserial/pyserial>. Acesso em: 15 jul. 2019.
- PYTHON. Welcome to Python.org. Disponível em: <https://www.python.org/>. Acesso em: 9 jul. 2019.
- SOLOMAN, S. **Sensores e Sistemas de Controle na Indústria**. 2. ed. LTC, 2012.
- STEVAN JR., S. L.; SILVA, R. A. **Automação e Instrumentação Industrial com Arduino – Teoria e Projetos**. São Paulo: Érica, 2015.

Unidade 3

Os dados na nuvem e a IoT

Convite ao estudo

A computação em nuvem é uma das grandes aliadas no desenvolvimento de aplicações IoT robustas e eficientes. É por meio desses sistemas em nuvem que conseguimos obter dados, visualizá-los e analisá-los de maneira remota. Isso aumenta a produtividade e proporciona grande versatilidade às aplicações.

Nesta terceira unidade, você será apresentado a uma ferramenta chamada Raspberry Pi (Rpi), que possibilita desenvolver aplicações de IoT integradas à arquitetura em nuvem. O Rpi é uma placa de prototipagem com características similares a um microcomputador. Com ela, você conseguirá utilizar sensores e atuadores conectados em uma mesma placa, além de ser possível visualizar os dados coletados diretamente em um monitor ou aparelho de televisão. Outra característica do Rpi é a possibilidade de integrar essa plataforma diretamente com a nuvem de dados. Então, como você deve ter percebido, essa unidade é de grande importância para você implementar a IoT para solução de problemas reais integrados a sistemas de computação em nuvem.

Nesta unidade, você conhecerá, além do Rpi, algumas outras plataformas que possibilitam integrar sensores e atuadores, além de realizar o processamento local dos dados. Essas plataformas são conhecidas como Computadores de Uma Placa Única (SBC do inglês *Single Board Computers*). Um destaque especial para o SBC Rpi será dado nesta unidade. Serão apresentados exemplos de como utilizá-la como plataforma para a IoT. Também, nesta unidade, você conhecerá os conceitos de computação em nuvem aplicados à IoT, verá como implementar esses conceitos por meio do Rpi e integrar esse SBC com as plataformas de visualização de dados em nuvem. A unidade termina mostrando como podemos armazenar e analisar os dados coletados pela aplicação. Essa análise é fundamental, pois é por meio dela que conseguimos gerar o conhecimento necessário sobre o ambiente em que a aplicação IoT foi desenvolvida.

Ao término desta unidade você será capaz de aplicar os conceitos aprendidos em unidades anteriores na construção de aplicações IoT utilizando o Rpi. Você conhecerá todo o fluxo de desenvolvimento de uma aplicação IoT por meio do SBC Rpi, desde a utilização do Rpi como plataforma de

integração de sensores e atuadores até a visualização e o tratamento desses dados em sistemas de computação em nuvem.

Como você já possui um grande conhecimento sobre aplicações IoT, você acaba de ser contratado por uma empresa de instrumentação. Essa empresa desenvolve sistemas e instrumentos inteligentes capazes de sentir (monitorar) e interagir (modificar) com o ambiente de instalação da aplicação. Os instrumentos desenvolvidos por essa empresa são conectados à internet, o que possibilita a visualização e a análise desses dados via plataformas em nuvem. Assim, com a instalação desses equipamentos, os clientes conseguem gerar os dados sobre o ambiente monitorado, analisar e retirar informações sobre esse ambiente e, a partir do conhecimento gerado, agir ou modificar o ambiente.

Essa empresa foi contratada para desenvolver um sistema inteligente capaz monitorar as variações climáticas e gerar informações que possibilitem um melhor aproveitamento da produção. Esse serviço de instrumentação inteligente foi contratado por uma grande fazenda automatizada de produção de soja, que possui uma estrutura em que toda a área plantada tem acesso à internet. O objetivo é conseguir monitorar algumas variáveis climáticas como temperatura ambiente, velocidade do vento, radiação solar, pressão atmosférica, umidade do ar e altitude em vários pontos da plantação. Além disso, desejam que imagens sejam retiradas de várias amostras das plantas, a fim de ser possível identificar a existência de pragas e mapear toda a plantação. Assim é possível gerar insumos para a tomada de decisão e aumentar os lucros da produção. A integração, visualização e análise dos dados devem ser realizadas em plataformas de programação em nuvem. Ao final dessa tarefa, os diretores da empresa desejam que seja entregue um relatório sobre o projeto e um protótipo utilizando a plataforma Rpi que seja capaz de desempenhar o monitoramento e a integração desses dados com a nuvem.

Bons estudos!

Seção 1

O Raspberry Pi e a IoT

Diálogo aberto

A grande difusão das aplicações IoT ocorreu, em parte, devido à existência dos SBC (do inglês, *Single Board Computer*). As plataformas de SBC são computadores que possuem unidades de processamento, memória e capacidade de integração de sensores e atuadores em uma única placa de circuito impresso (BIRMINGHAM; SIEWIOREK ,1987). Esses SBC estão em várias aplicações da IoT, desempenhando diferentes funções e auxiliando na resolução de problemas (FEDRIZZI; SORAIA, 2015; ISIKDAG, 2015).

Você foi contratado por uma grande empresa de instrumentação que desenvolve sistemas inteligentes integrados à internet e capazes de obter dados, enviá-los para plataformas em nuvem, analisá-los e gerar informações para uma tomada correta de decisão. Você é o chefe de desenvolvimento de uma aplicação que deve monitorar o microclima e a saúde de uma grande plantação de soja. A primeira etapa do trabalho a ser realizado consiste em apresentar um relatório contendo alguns sensores utilizados para monitorar o microclima da plantação e como alguns algoritmos de aprendizado de máquina podem ser empregados para gerar informações relevantes sobre a aplicação e atender aos requisitos impostos pelos clientes.

Na parte inicial desta seção, você será apresentado ao Rpi e a outros SBC, logo após, você vai aprender como o Rpi pode ser integração à IoT. Você verá, ainda, como conectar sensores e atuadores ao Rpi e como a programação desse dispositivo é realizada para permitir o monitoramento e modificação do ambiente. Assim, você terá conhecimento sobre todos os elementos necessários para construir a primeira etapa do projeto de desenvolvimento do sistema inteligente. Além disso, vai compreender as questões que envolvem o Rpi, aprenderá como ligar os sensores a ele e como desenvolver programas para a integração desses sensores/atuadores. Conhecerá, também, algumas aplicações de algoritmos de aprendizado de máquina que podem ser aplicados para a construção de aplicações inteligentes e eficientes para a IoT.

Os diretores da fazenda de produção de soja estão ansiosos com a possibilidade de automatizar o monitoramento climático da plantação. Eles contam com você para gerar dados confiáveis que possam ser utilizados como base para a tomada de decisão.

Não pode faltar

Os Single-Board Computer (SBC)

Os dispositivos (ou objetos inteligentes) que compõem a Internet das Coisas representam as unidades fundamentais de construção desse complexo sistema de comunicação e interação com o ambiente. A grande importância desses objetos inteligentes para a Internet das Coisas é o fato de possibilitarem que coisas do nosso cotidiano sejam capazes de sentir/interagir com o ambiente e se comunicarem entre si e com os seres humanos por meio da internet (FORTINO; TRUNFIO, 2014). Quando os objetos físicos de nosso dia a dia são dotados dessa capacidade de comunicação por meio da internet e da interação com o ambiente, torna-se possível criar as bases para o desenvolvimento da IoT.

Com o desenvolvimento dos sistemas embarcados, miniaturização e barateamento dos custos de produção de componentes eletrônicos, várias plataformas de prototipagem passaram a ser desenvolvidas. Quando correspondem a computadores que possuem processador e memória em uma mesma placa de circuito, elas são chamadas de *Single-Board Computer* (SBC) (ISIKDAG, 2015(a)). Os SBC, normalmente, possuem conexões de entrada e saída integradas a outros componentes em uma mesma placa de circuito. Ter todos os componentes necessários para construção de uma aplicação em uma única placa proporciona maior comodidade e facilidade ao desenvolvimento de aplicações.

Os SBC são utilizados, normalmente, como placas de prototipagem, ou seja, são empregados como prova de conceito e realização de experimentos. Existem diferentes modelos de SBC, como o Raspberry Pi (Rpi) (RASPBERRY PI, [s.d.]); o Arduino (WHAT IS ..., [s.d.]); o Beagleboard (BEAGLEBOARD. ORG, 2019) e o Udoos (UDOOS, [s.d.]). A Figura 3.1 apresenta uma imagem ilustrativa dos SBC citados.

Figura 3.1 | Exemplo de diferentes SBC utilizados na IoT



a) Raspberry Pi 3



b) Arduino Uno R3



c) BeagleBone Black



d) Udooboard

Fonte: adaptada de Maksimović et al. (2014).

O Rpi é um SBC introduzido em 2012, do tamanho de um cartão de crédito, que possui características semelhantes a de um computador pessoal (MAKSIMOVIC et al., 2014). Ele foi construído para ser uma plataforma educacional que pudesse ser utilizada no ensino da Ciência da Computação nas escolas (RASPBERRY PI, 2016). O Rpi possui um sistema operacional, saída HDMI (*High-Definition Multimedia Interface*) e conexões *Universal Serial Bus* (USB) para utilização de teclado e mouse. Além disso, ele pode realizar conexão com a internet por meio de cabo Ethernet ou Wi-Fi e apresenta vários pinos de entrada e saída de dados (MAKSIMOVIC et al., 2014). Todas essas características, combinadas com o baixo preço de aquisição, tornam o Rpi um excelente candidato a ser empregado no desenvolvimento de aplicações IoT. Graças aos SBC com maior facilidade de programação, temos uma maior divulgação e construção de projetos utilizando os conceitos da IoT.

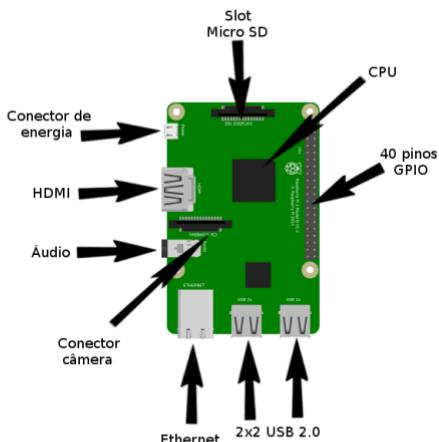
Exemplificando

Atualmente, existe uma grande quantidade de fabricantes investindo na construção de SBC. Grandes empresas de hardware estão investindo no desenvolvimento de placas de prototipagem. Um exemplo é o Galileo (INTEL, [s.d.]), que utiliza os conceitos empregados no arduino para criar um produto específico da marca. Isso mostra que mesmo grandes empresas enxergam o potencial dessas placas de prototipagem.

O Rpi possui pinos de entrada/saída de dados, que são conhecidos como *General Purpose Input/Output* (GPIO). São esses pinos que possibilitam a conexão de sensores/atuadores. A Figura 3.2 ilustra o modelo Rpi 3 B+ e a localização dos principais componentes existentes nessa placa. Observando a Figura 3.2, é possível constatar que em apenas uma placa de Circuito

Impresso (CI) temos vários dos componentes necessários para desenvolver aplicações de IoT. Como o Rpi também apresenta a possibilidade de conexão com a internet sem a necessidade de adicionar circuitos extras, realizar a troca de mensagens com a nuvem de dados torna-se uma tarefa menos trabalhosa. Tudo isso a um custo acessível. Portanto, essas são as características que tornam o Rpi uma das mais interessantes ferramentas de construção e divulgação de projetos em IoT.

Figura 3.2 | Rpi e seus principais componentes



Fonte: elaborada pelo autor.

Assimile

Os SBC podem ser vistos como uma parte fundamental para o desenvolvimento de aplicações IoT. Entretanto, cada uma das placas construídas apresenta características próprias. A escolha sobre qual deve ser utilizada para o desenvolvimento de seus projetos deve ser realizada levando-se em consideração as peculiaridades do problema que se deseja resolver por meio da aplicação da IoT.

O Sistema Operacional (SO) utilizado para o Rpi é o Raspbian, que é um SO de código aberto, baseado em Debian (uma distribuição do Linux) (HARRINGTON, 2015). O Raspbian foi desenvolvido especialmente para o Rpi, portanto, todas as funcionalidades dessa plataforma podem ser alcançadas utilizando esse SO. O Raspbian, assim como outras distribuições baseadas em Linux, apresenta, nativamente, compatibilidade com a linguagem Python. Com a utilização das bibliotecas desenvolvidas nessa linguagem, podemos conectar vários sensores e atuadores aos GPIO do Rpi.

sem a necessidade de conhecer, profundamente, disciplinas como elétrica e eletrônica. Essa também é uma das vantagens de utilização do Rpi como plataforma para a IoT.

Pesquise mais

Assista ao vídeo que mostra a instalação do Raspbian, o Sistema operacional do Raspberry PI 3, por meio do ZIP chamado de NOOBS.

RASPBERRY PI 3 - instalando sistema operacional. [S.I.: s.n.], 2016. 1 vídeo (38:53 min). Publicado pelo canal Projeto Root. [Youtube](#), 2016.

Para conectarmos sensores/atuadores à placa Rpi, primeiramente, é necessário conhecer a localização e as funções de cada um dos pinos GPIO. Isso será necessário, pois cada um deles possui funções específicas de entrada ou saída de dados. Além disso, determinados pinos apresentam características especiais, por exemplo, os pinos 14 e 15. Esses dois pinos são utilizados para comunicação serial UART (*Universal Asynchronous Receiver/Transmitter*). A Figura 3.3 apresenta algumas funções realizadas por pinos específicos do Rpi.

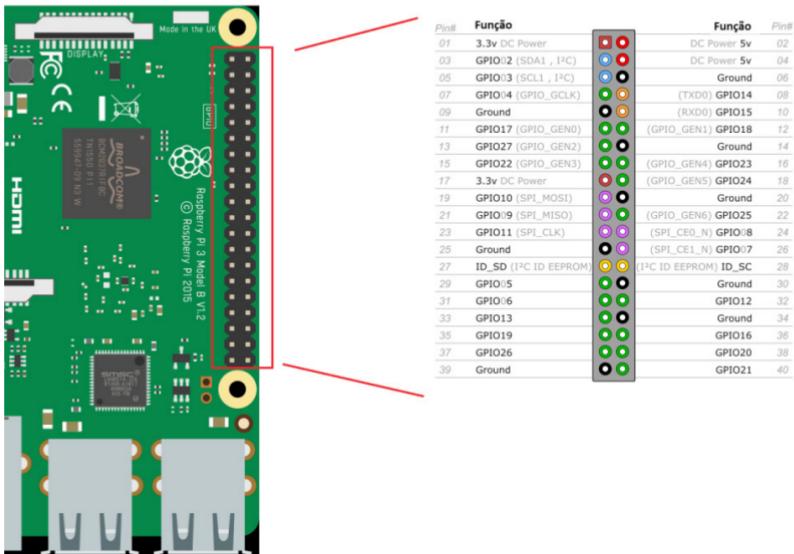
Figura 3.3 | Algumas funções específicas dos pino do Rpi

Pino	NOME	NOME	Pino
01	3.3v DC Power	DC Power	02
03	GPIO02 (SDA1 , I ² C)	DC Power	04
05	GPIO03 (SCL1 , I ² C)	Ground	06
07	GPIO04 (GPIO_GCLK)	(TXD0) GPIO14	08
09	Ground	(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)	(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)	Ground	14
15	GPIO22 (GPIO_GEN3)	(GPIO_GEN4) GPIO23	16
17	3.3v DC Power	(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)	Ground	20
21	GPIO09 (SPI_MISO)	(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)	(SPI_CE0_N) GPIO08	24
25	Ground	(SPI_CE1_N) GPIO07	26
27	ID_SD (I ² C ID EEPROM)	(I ² C ID EEPROM) ID_SC	28
29	GPIO05	Ground	30
31	GPIO06	GPIO12	32
33	GPIO13	Ground	34
35	GPIO19	GPIO16	36
37	GPIO26	GPIO20	38
39	Ground	GPIO21	40

Fonte: adaptada de Element14 (2016).

UART é um protocolo de comunicação serial em que cada dado é transmitido bit a bit (HOCKER III, 1997). O pino 14 apresenta a função de TX (Transmissor) para o protocolo UART, já o pino 15 possui a função de RX (Receptor). Com o UART é possível estabelecer a comunicação entre dois dispositivos utilizando-se apenas 2 cabos. Assim, podemos utilizar um cabo para ligar o pino 14 de um Rpi ao pino 15 de outro Rpi e o segundo cabo conectado de forma inversa. Dessa forma, conseguimos facilmente realizar a comunicação entre esses dois Rpi.

Figura 3.4 | Localização dos pinos GPIO para o Rpi 3 B+



Fonte: adaptada de Element14 (2016).

Implantando dispositivos IoT com o Raspberry Pi

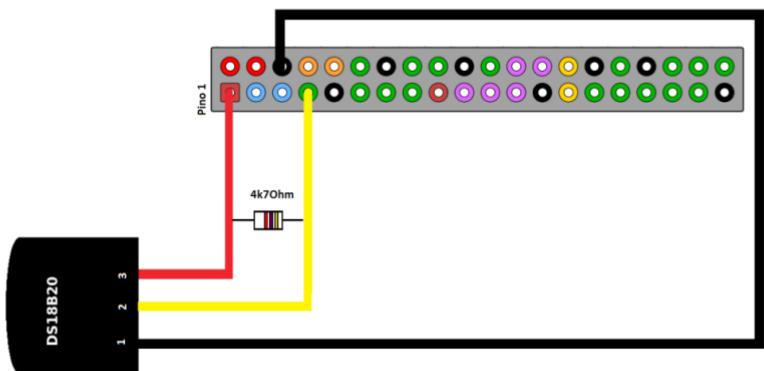
Com a utilização do Rpi, podemos construir instrumentos inteligentes de medição. Para a construção de um instrumento capaz de ler a temperatura ambiente e enviar esses dados por meio da internet para alguma plataforma em nuvem, por exemplo, pode ser utilizado o Rpi e o sensor DS18B20.

O DS18B20 é um sensor de temperatura dotado de um circuito integrado que possibilita enviar os dados lidos para o Rpi através de apenas 1 fio (SEMICONDUCTOR, 2017). Como cada um dos DS18B20 possui um endereço físico, é possível usar vários desses sensores utilizando apenas 1 barramento do Rpi. Com o emprego do Rpi, é possível converter um pequeno sensor de temperatura em um instrumento conectado à internet. Ao

conectar esse dispositivo à internet e possibilitar a comunicação com outros elementos, estamos começando a desenvolver as bases para a construção de uma aplicação IoT.

A Figura 3.5 apresenta o sensor DS18B20 e a indicação sobre cada um dos 3 fios que ele possui. Nessa mesma figura é possível ver como o DS18B20 pode ser conectado aos GPIO do Rpi. Após conectar esse sensor, devemos construir o programa de computador que deverá indicar como os dados de temperatura devem ser coletados. A linguagem Python é nossa grande aliada nessa tarefa. A Figura 3.6 apresenta um exemplo de código escrito em Python para leitura da temperatura através do sensor DS18B20.

Figura 3.5 | Esquema de ligação do DS18B20 aos GPIO do Rpi



Fonte: elaborada pelo autor.

Figura 3.6 | Exemplo de código escrito em Python para a leitura do sensor de temperatura

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
import os
import glob
import time

os.system('modprobe w1-gpio') #ativar a leitura através de 1 fio
os.system('modprobe w1-therm') #ativar a leitura para o termômetro

base_dir = '/sys/bus/w1/devices/' #caminho para a leitura da temperatura
device_folder = glob.glob(base_dir + '28*')[0] #obtem o serial do sensor de temperatura
device_file = device_folder + '/w1_slave' #completa o caminho para leitura (1 fio como escravo)

#função para ler o arquivo gerado após cada leitura
def le_temp_sem_tratamento():
    arq = open(device_file, 'r') #abre o arquivo que contém a leitura do sensor
    linhas = arq.readlines() #lê as linhas desse arquivo
    arq.close() #fecha o arquivo
    return linhas #retorna as linhas lidas

#função para realizar a leitura da temperatura
def le_temp():
    linhas = le_temp_sem_tratamento() #recebe as linhas do arquivo
    while linhas[0].strip()[-3:] != 'YES': #continua lendo até encontrar uma leitura válida
        time.sleep(0.2)
    igual = linhas[0].find('t=')
    if igual != -1:
        temp = linhas[0][igual+2:]
        return temp
```

```

linhas = le_temp_sem_tratamento()
equals_pos = linhas[1].find('t=')
if equals_pos != -1: #caso a leitura seja válida
    temp_string = linhas[1][equals_pos+2:] # retira a parte correspondente a temperatura
    temp_c = float(temp_string) / 1000.0 #temperatura em graus Celsius
    temp_f = temp_c * 9.0 / 5.0 + 32.0 #temperatura em Fahrenheit
    return temp_c, temp_f

#Realiza a leitura em loop infinito
while True:
    print(le_temp())
    time.sleep(1)

```

Fonte: adaptada de Basics (2016).

Após conectar o DS18B20 ao Rpi, utilizando bibliotecas, e construir o programa em Python que seja capaz de ler os dados recebidos do DS18B20, chegou a hora de possibilitar que os dados sejam acessados por meio da internet. Para disponibilizar essas medições dentro de uma rede local vamos, novamente, utilizar a linguagem Python. O acesso aos dados por meio da internet se tornará possível utilizando o microframework web Flask (THE PALLETS PROJECTS, [s.d.]).

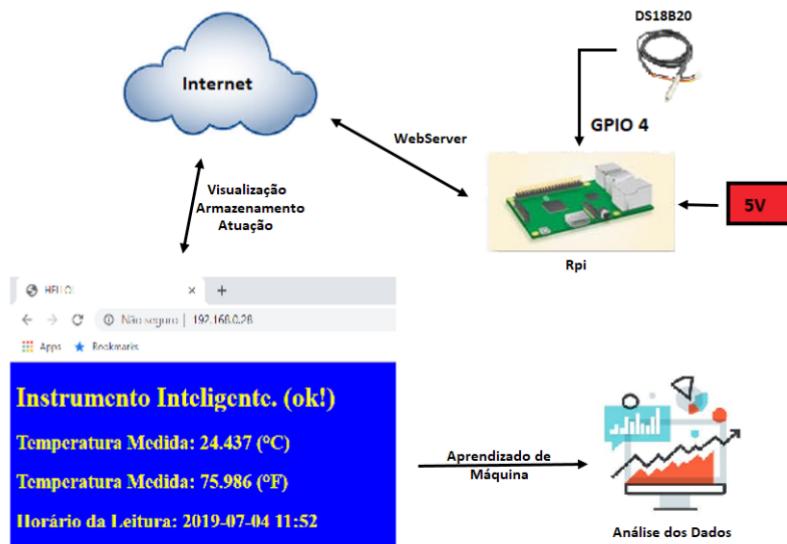
Por meio do Flask é possível construir aplicações web de maneira mais rápida e descomplicada (GRINBERG, 2018). Assim, podemos construir um programa que seja capaz de enviar e receber dados por meio da internet.



Acesse o QR Code ou o link e veja no código-fonte um exemplo de programa construído a partir da linguagem Python e do microframework Flask para possibilitar a comunicação com Rpi por meio da internet.
http://cm-kls-content.s3.amazonaws.com/ebook/embed/qr-code/2019-2/internet%20das%20coisas/u3/s1/int_coi_u3_s1_QRCode_bruto.pdf

Após essas etapas, pode ser construído o termômetro capaz de realizar medidas e trocar mensagens por meio da internet. A Figura 3.7 mostra um esquema de construção de um instrumento inteligente. Para a criação de um instrumento/objeto inteligente é necessário que exista a capacidade de comunicação e interação com o ambiente. Esse sensor, além de conseguir monitorar o ambiente, terá a possibilidade de interagir e trocar mensagens com outros objetos e com os usuários da aplicação.

Figura 3.7 | Construção de um instrumento/objeto inteligente



Fonte: adaptada de Mjrobot (2018).

O Raspberry Pi e o aprendizado de máquina para a IoT

Os instrumentos inteligentes são responsáveis por grande parte dos dados gerados atualmente. A quantidade de dados produzida todos os dias cresce a um ritmo nunca antes visto. Com mais de 50 bilhões de dispositivos conectados à internet (EMMERSON, 2010), as aplicações IoT são um dos principais elementos que contribuem para o incremento dessa quantidade de dados. Esse grande volume de dados, gerados a uma velocidade muito alta e provindos de uma enorme variedade de aplicações e dispositivos, ocasionou o surgimento do Big Data (KITCHIN; MCARDLE, 2016). Essas características representam os 3 V's do Big Data (Volume, Velocidade e Variedade). Para analisar e retirar conhecimento de toda essa massa de dados, serão necessários métodos computacionais que possam ajustar, agrupar e extrair informações relevantes de maneira automática. É esse conhecimento gerado que possibilita ao usuário prever e controlar o ambiente de maneira mais precisa.

Um dos grandes responsáveis por possibilitar a extração de conhecimento desses dados é o aprendizado de máquina (Machine Learning - ML), que é a área de pesquisa que investiga o modo de proporcionar aos computadores a habilidade de aprender sem a necessidade de serem explicitamente programados (SAMUEL, 1959), trata-se de uma subárea da inteligência artificial. São os algoritmos de ML que possibilitam aos computadores a capacidade de realizar previsões e classificar de maneira automática um conjunto de dados.

Atualmente, encontramos aplicações de ML em quase todas as áreas do conhecimento. Em seu dia a dia, provavelmente, você deve utilizar vários serviços que empregam algoritmos de ML. Nos smartphones, por exemplo, existem os assistentes pessoais virtuais, que respondem a comandos de voz, realizam pesquisas, fazem chamadas telefônicas, leem recado, enviam mensagens e uma série de outras funções apenas seguindo os comandos enviados pelos usuários. Cabe ressaltar que tudo isso é feito em a necessidade de digitar qualquer tipo de palavra (LÓPEZ; QUESADS; GUERREIRO, 2017).

Quando você recebe um e-mail e ele vai direto para a caixa de SPAM, esta também é uma aplicação de técnicas de ML. Os algoritmos desenvolvidos são utilizados para reconhecer as palavras ou combinações de frases que mais se assemelham a e-mails não solicitados ou maliciosos, classificando-os como SPAM (JINDAL; LIU, 2007).

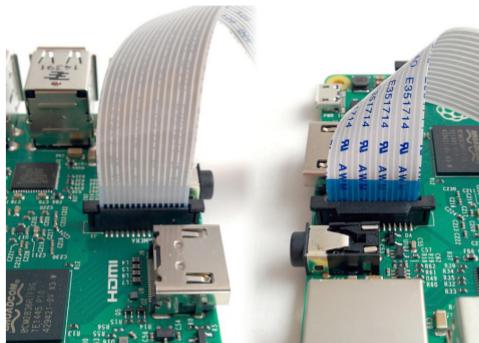
Ao utilizar as ferramentas de busca na internet (CROFT; METZLER; STROHMAN, 2010) ou até mesmo fazer uso do corretor ortográfico de seu editor de texto (BRILL; MOORE, 2000), você também está utilizando algum modelo de aprendizado de máquina. Quando corrigimos um retorno de busca que não estávamos esperando ou retificamos alguma sugestão do corretor ortográfico, estamos contribuindo para que esse algoritmo possa “aprender” com esses erros cometidos e, assim, melhorar o próprio desempenho.

Outra aplicação bastante interessante desses algoritmos é a identificação facial ou de objetos em uma imagem (REDMON *et al.*, 2016). Essa tecnologia é utilizada em veículos autônomos para identificar placas, faixas, outros veículos e pedestres (MENZE; GEIGER, 2015). A identificação facial é empregada em câmeras de segurança e de pessoas em redes sociais e em câmeras fotográficas (JAIN; LI, 2011).

Como é possível observar, os algoritmos de ML são grandes aliados na melhoria dos processos e produtos existentes, no aproveitamento de recursos e na melhoria da qualidade de vida das pessoas. Para a IoT, esses algoritmos também se apresentam como grandes parceiros para o desenvolvimento de aplicações. O Rpi possui características similares a um computador pessoal, assim temos a possibilidade de implementarmos sistemas inteligentes utilizando o aprendizado de máquina. O Rpi pode utilizar o ML, por exemplo, para realizar o reconhecimento facial (SENTHILKUMAR; GOPALAKRISHNAN; KUMAR, 2014). Como no Rpi existe a possibilidade de se adicionar uma câmera diretamente à placa de circuito, aplicações que utilizam algoritmos de processamento de imagens encontram nele uma ferramenta bastante poderosa. A Figura 3.7 apresenta a conexão da câmera ao Rpi. Após a conexão, pode ser utilizada, por exemplo, a biblioteca *Open Source Computer Vision Library* (OpenCV) para realizar a detecção automática de faces. O OpenCV utiliza algoritmos de classificação para identificar e marcar a região da

imagem que corresponde ao rosto humano. A Figura 3.8, apresenta a programação do Rpi para realizar a detecção facial.

Figura 3.8 | Conexão da câmera ao Rpi



Fonte: Foundation (2016)

Figura 3.9 | Código para a detecção facial

```
1. # -*- coding: utf-8 -*-
2. #
3. #definição das bibliotecas
4.
5. import cv2 # biblioteca openCV - utilizada para os
   processos que envolvem o tratamento de imagem
6. import matplotlib.pyplot as plt # mostrar a imagem no
   tamanho desejado
7.
8. #
9. #Realiza a classificação
10.
11.# Criação do objeto CascadeClassifier
12.# É esse classificador que encontrar a porção da imagem
   que corresponde às faces
13.face_detection = cv2.CascadeClassifier("haarcascade_
   frontalface_default.xml") #arquivo xml que contém o
   treinamento do HaaR
14.
15.# Carregando a imagem
16.img = cv2.imread("einsten.jpg") # pode ser escolhida
   qualquer imagem para realizar a identificação
17.# Lendo a imagem em uma escala de cinza
```

```

18. gray_img      =      cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    #carrega a imagem em escala de cinza para facilitar a
    classificação

19.

20. # procurando pelas coordenadas da imagem

21. faces      =      face_detection.detectMultiScale(gray_img,
    scaleFactor = 1.05, minNeighbors=5) #

22.

23. for x,y,w,h in faces:          #identifica as coordenadas
    da foto e desenha o retângulo

24. img = cv2.rectangle(img, (x,y), (x+w,y+h),(0,255,0),3)
    #desenha o retângulo ao redor da imagem

25.

26. #-----
-----
```

27. # Realiza o plot das imagens

28. plt.figure(figsize=(20,10)) #define o tamanho da imagem a ser plotada

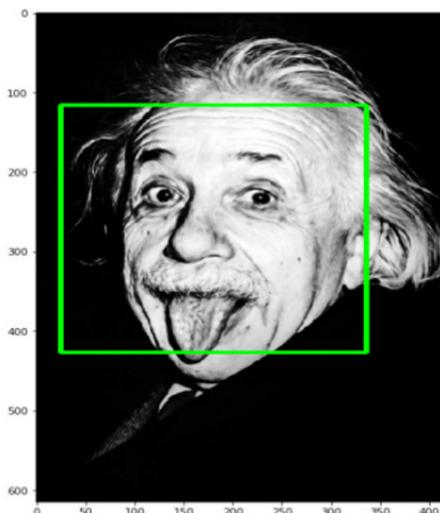
29. plt.imshow(img) # mostra a imagem

30.

31. cv2.waitKey(0) # espera a entrada de alguma tecla para fechar a imagem criada

32.

33. cv2.destroyAllWindows() #fecha a imagem



Fonte: elaborada pelo autor.

Essas características também são úteis para a agricultura (ZHONG *et al.*, 2018), pois o tamanho reduzido, o baixo custo, a existência dos GPIO e a capacidade de conexão à internet auxiliam na implantação desse dispositivo em vários locais de uma lavoura. Assim, o Rpi apresenta-se como uma ótima ferramenta de integração de sensores/atuadores que pode ser utilizada como meio de implementação dos algoritmos de ML.

Chegamos ao final desta seção. Você, certamente, aprendeu um pouco mais sobre a IoT, SBC e como esses elementos podem ser utilizados em conjunto com os algoritmos de ML a fim de gerar aplicações mais robustas e que sejam mais eficientes.

Sem medo de errar

O desenvolvimento e o barateamento dos SBC têm papel fundamental no crescimento e disseminação da IoT. A adoção dos SBC como plataformas para a construção de aplicações possibilita que usuários com diferentes níveis de conhecimento em elétrica e eletrônica possam criar projetos que utilizem os conceitos da IoT. Assim, parte do sucesso da IoT decorre das vantagens de utilização dos SBC.

Você está trabalhando em uma empresa de instrumentação, que projeta e constrói instrumentos de medição inteligentes capazes de monitorar e interagir com o ambiente e com outros equipamentos por meio da internet. Uma de suas demandas é desenvolver um equipamento que seja capaz de monitorar o microclima de uma grande plantação de soja. Esse monitoramento do microclima é realizado por meio do acompanhamento de algumas variáveis climáticas locais, como temperatura, umidade do ar, velocidade do vento, altitude. Para isso, ao final de todo o projeto, deve ser construído um protótipo desse instrumento utilizando o Rpi. Na primeira fase do trabalho, você deve entregar um relatório contendo a indicação de alguns sensores a serem utilizados para monitorar a plantação e indicar como os algoritmos de ML podem ser úteis para o monitoramento dessa plantação.

O Quadro 3.1 sintetiza alguns sensores e as variáveis monitoradas por eles em uma estação de acompanhamento climático. A resposta presente nesse quadro não é a única possível.

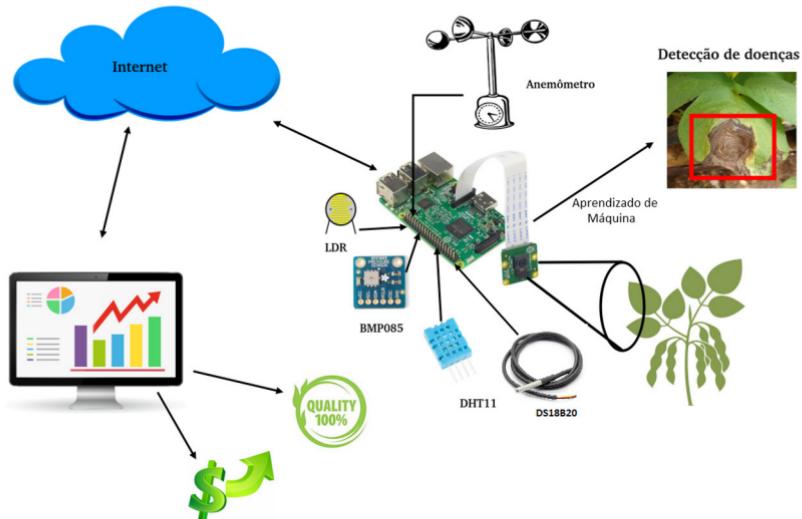
Quadro 3.1 | Sugestão de resposta ao exercício

Sensor	Variável a ser monitorada
BMP085	Pressão barométrica e altitude
DS18B20	Temperatura atmosférica
Anemômetro	Velocidade do vento
LDR	Radiação solar
DHT11	Umidade do ar

Fonte: elaborado pelo autor.

Como aplicação de machine learning utilizando o Rpi, é possível empregar os algoritmos de detecção de objetos em imagens, que podem ser utilizados para identificar quais plantas estão sendo atacadas por pragas de maneira automática. Após esses algoritmos serem treinados, pode-se utilizar uma câmera do Rpi e capturar imagens, periodicamente, de diferentes regiões da cultura. Assim, é possível que as plantas possam ser classificadas de maneira automática como pertencentes ao grupo de plantas sadias ou não. A Figura 3.10 apresenta o esquema de utilização do Rpi para detectar plantas infetadas e monitorar do microclima da região.

Figura 3.10 | Esquema de uma aplicação utilizando o Rpi para monitoramento de lavoura



Fonte: elaborada pelo autor.

Raspberry Pi como porteiro inteligente

Você, ao chegar em casa depois de um dia longo de trabalho, percebeu que o porteiro de seu prédio não estava na guarita de segurança. Você foi até a portaria e começou a chamá-lo, mas só após alguns minutos é que conseguiu entrar em seu apartamento. Como você conhece as vantagens de se utilizar a IoT para resolver problemas reais, decidiu propor, durante a reunião de condomínio, que fosse testado um porteiro eletrônico digital inteligente. O síndico é um engenheiro aposentado e que gosta de aprender novas tecnologias, logo ficou bastante animado com essa possibilidade. Após a reunião, todos concordaram que essa seria uma possível solução para a maior segurança e comodidade no condomínio.

Como todos sabem do conhecimento que você possui e da sua competência em resolver problemas com a utilização da IoT, você foi o encarregado de propor um protótipo para ser utilizado em testes pelo condomínio. Ficou decidido após a reunião que esse sistema deve conseguir identificar quando alguém está próximo, automaticamente ligar uma câmera e retirar uma foto dessa pessoa. Além disso, caso o indivíduo que deseja entrar seja um morador do condomínio, deve permitir a entrada, caso contrário deve acionar o porteiro. O síndico do condomínio espera que você entregue um relatório contendo os sensores e atuadores a serem utilizados e como pode ser realizado o reconhecimento facial dos moradores.

Resolução da situação-problema

Como essa aplicação requer a criação de um objeto capaz de monitorar e interagir com o ambiente, devem ser adicionados sensores e atuadores ao Rpi. O Quadro 3.2 apresenta os sensores/atuadores a serem utilizados e as respectivas funções.

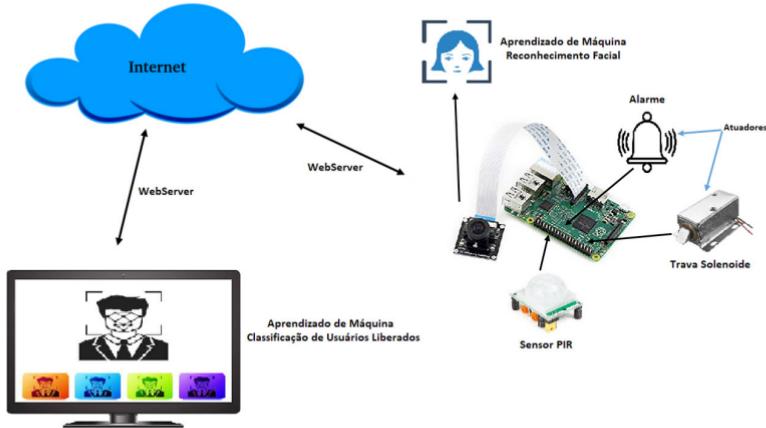
Quadro 3.2 | Sugestão de resposta para o problema do Rpi como porteiro inteligente

Sensor/atuador	Função
Sensor PIR	Verificar a presença de indivíduos nas proximidades da portaria
Chave solenoide	Abertura da porta
Alarme	Informar o porteiro sobre a chegada de alguma pessoa

Fonte: elaborado pelo autor.

Para o reconhecimento dos moradores, é necessário que seja realizado um cadastro prévio de todos os residentes no condomínio. Assim, utilizando os algoritmos de ML empregados para o reconhecimento facial, é possível classificar se o indivíduo que se dirigiu à portaria pertence ao grupo de moradores ou não. Caso pertença, deve ser gerado o acionamento da chave solenoide, caso contrário, deve ser enviado um alarme diretamente ao porteiro. A Figura 3.11 apresenta um esquema contendo os principais componentes desse sistema e a relação entre eles.

Figura 3.11 | Sugestão de esquema para construção do controle de acesso inteligente



Fonte: elaborada pelo autor.

Faça valer a pena

1. Os SBC possuem grande contribuição para a disseminação da IoT. Os “maker”, como são conhecidos os usuários que utilizam os SBC para a construção de aplicações reais, surgiram com o aumento da presença das placas de prototipagem de código aberto. Por possuírem todos os componentes em uma mesma placa de circuito e disponibilidade dos GPIO, os SBC são utilizados em várias aplicações. Entretanto, em situações em que questões de segurança, por exemplo, são essenciais, não se deve empregar os SBC como aplicação final. Abaixo estão listadas algumas aplicações em que podemos utilizar o Rpi como produto final.

- I. Aplicações de domótica.
- II. Construção de veículos autônomos.

III. Monitoramento de condições climáticas para um jardim.

IV. Desenvolvimento da Indústria 4.0.

Com base no contexto apresentado, é correto o que se afirma em:

- a. I, II e III, apenas.
- b. IV, apenas.
- c. I e III, apenas.
- d. III e IV, apenas.
- e. I e IV, apenas.

2. Os SBC são plataformas muito utilizadas para monitoramento e interação com o ambiente via internet. Dentre os SBC mais conhecidos temos o Rpi. Os itens abaixo indicam as características do Rpi que o tornam uma excelente plataforma de prototipagem para o desenvolvimento da IoT.

- I. Grande quantidade de GPIO.
- II. Processamento superior a um computador atual.
- III. Compatibilidade com a linguagem Python.
- IV. Wi-Fi e Ethernet integrados.
- V. SO (Raspbian) baseado no Windows.

Considerando as características do Rpi, é correto o que se afirma em:

- a. I, II e III, apenas.
- b. I, II e V, apenas.
- c. III e V, apenas.
- d. I, II e IV, apenas.
- e. IV e V, apenas.

3. Algoritmos de aprendizado de máquina são essenciais para o desenvolvimento da IoT (LI; OTA; DONG, 2018). A utilização desses algoritmos em plataformas do tipo SBC representa uma grande oportunidade para a criação de novas aplicações para a IoT e melhoria de processos já existentes (LI; OTA; DONG, 2018). Entretanto, aplicações que exigem grande capacidade de processamento para a utilização do aprendizado de máquina não devem ser utilizadas, diretamente nos SBC. Isso ocorre, pois esses dispositivos possuem sistemas de processamento e armazenamento inferiores aos

computadores pessoais, por exemplo. Abaixo, estão listadas algumas aplicações em que as questões de desempenho não impedem a utilização dos SBC em conjunto com algoritmos de aprendizado de máquina.

- I. Aplicações em veículos autônomos.
- II. Aplicações para o monitoramento de pacientes.
- III. Aplicações de Domótica.
- IV. Controle de acesso via reconhecimento facial.
- V. Controle de acesso via impressão digital.

Indique qual a alternativa apresenta a combinação correta das afirmativas anteriores.

- a. I e IV, apenas.
- b. II, III, IV e V, apenas.
- c. I, III e IV, apenas.
- d. II, III e IV, apenas.
- e. I, II, III e IV, apenas.

Seção 2

Computação em nuvem e a IoT

Diálogo aberto

Você já parou para pensar no quanto a computação em nuvem modificou vários paradigmas que possuímos? E sobre todas as possibilidades que ela gerou? Antigamente, a única forma existente para construir programas, visualizar dados e processá-los era por meio dos nossos computadores pessoais. Todos esses programas deveriam estar instalados em nossos Hard Disk (HD) e apenas utilizarem a capacidade de processamento local do computador. Hoje, conseguimos realizar todas essas tarefas de armazenamento, visualização e processamento utilizando recursos disponíveis em máquinas espalhadas pelo mundo. Apenas precisamos ter uma conexão com a internet para enviarmos os comandos necessários e receber as respostas pedidas.

O projeto de construção de um sistema inteligente que possibilite o monitoramento do microclima da região da fazenda de soja está a todo vapor. Agora, você está na segunda etapa de elaboração do projeto e precisa indicar como os dados coletados devem ser integrados com as plataformas em nuvem. Os diretores esperam que nesta segunda etapa você entregue um relatório indicando como os dados coletados devem ser enviados para a nuvem e quais as possibilidades geradas pela utilização da programação em nuvem para essa aplicação. Esse relatório também será entregue para os clientes, assim, é necessário que você apresente o maior número de oportunidades que a nuvem de dados oferece para a melhoria da produção.

Para ajudá-lo nessa tarefa, nesta seção, vamos conhecer os conceitos da computação em nuvem, como eles são aplicados na IoT e alguns exemplos de utilização em aplicações reais. Assim, você terá todo o conhecimento necessário para escrever um relatório consistente e que impressione os diretores e os clientes.

Este é o momento de você convencer os clientes de que os instrumentos inteligentes conectados à internet podem melhorar a produtividade da lavoura, ajudar a aprimorar a qualidade do produto e aumentar os lucros. Você não vai deixar passar essa oportunidade. Os diretores contam com o seu trabalho para impressionar os clientes.

Não pode faltar

Conceitos de computação em nuvem aplicados à IoT

A utilização de serviços de rede para o compartilhamento de recursos já ocorre há muitos anos. A ideia de aproveitar as oportunidades que os serviços de rede oferecem para o compartilhamento de dados surgiu antes da própria internet. A primeira utilização do termo computação em nuvem foi feita em agosto de 2006, em uma conferência sobre estratégias para a melhoria de máquinas de buscas, foi mencionado um novo modelo de computação em que os serviços de dados e arquiteturas deveriam estar em servidores distribuídos (SCHMIDT, 2019).

A computação em nuvem pode ser vista como um modelo capaz de oferecer acesso ubíquo, conveniente e sob demanda a recursos computacionais compartilhados (MELL; GRANCE, 2011). Esses recursos podem ser, por exemplo, de armazenamento, compartilhamento de processamento e memória, aplicações e rede de dados. As principais características dos sistemas de computação em nuvem são: serviços sobre demanda, acesso ubíquo, elasticidade e disponibilidade de uma grande variedade de serviços (MELL; GRANCE, 2011).

Essas características da computação em nuvem vão ao encontro das premissas de utilização da Internet das Coisas: em “todo lugar”, a “todo tempo” e em “todas as coisas”. A computação possibilita oferecer o compartilhamento dos recursos gerados pelas aplicações IoT com vários usuários ao mesmo tempo por meio da internet (BISWAS; GIAFFREDA, 2014). Esses recursos podem ser, por exemplo, o acompanhamento de uma variável de interesse ou a modificação do ambiente através dos atuadores. Além disso, a utilização das plataformas em nuvem permite aos usuários visualizarem os dados de maneira remota utilizando diferentes dispositivos, tais como, smartphones, computadores pessoais ou tablets.

Segundo BOTTA *et al.* (2016), devido à limitação dos recursos presentes nos dispositivos (ou objetos inteligentes), utilizar os serviços de armazenamento e processamento disponibilizados pelas plataformas de computação em nuvem possibilita grandes oportunidades para o desenvolvimento das tecnologias.

As plataformas de computação em nuvem podem realizar processamento e armazenamento utilizando recursos compartilhados. Por outro lado, os serviços de computação em nuvem, ao utilizarem a IoT, podem contar com uma nova possibilidade de interação com o mundo real. Nesse sentido, a

IoT pode utilizar plataformas de computação em nuvem como uma camada intermediária entre as coisas (*things*) e os usuários da aplicação. Assim, a complexidade da implementação é reduzida, o que facilita a compreensão, interação e o desenvolvimento de diferentes aplicações.

O Quadro 3.3 apresenta um comparativo entre a IoT e a computação em nuvem. Pode-se observar que essas tecnologias apresentam características complementares. Logo, realizar a integração entre elas é uma estratégia que possibilita gerar aplicações mais eficientes (ATLAM *et al.*, 2017).

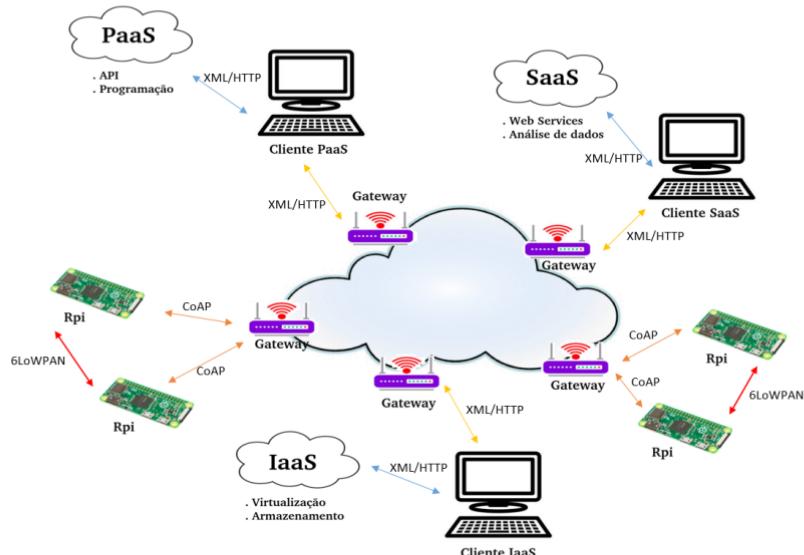
Quadro 3.3 | Comparação entre as tecnologias IoT e Computação em nuvem.

	IoT	Computação em Nuvem
Movimentação de Dados	Pervasiva	Centralizada
Visualização dos Dados	Limitada	Ubíqua
Componentes	Objetos do mundo real	Softwares (mundo virtual)
Capacidade Computacional	Limitada	Virtualmente ilimitada
Armazenamento	Limitado	Virtualmente ilimitado
Como Utiliza a Internet	Ponto de convergência	Meio para prover os serviços
Papel no Big Data	Fonte de dados	Gerenciamento de dados

Fonte: adaptado de Botta *et al.* (2016).

A junção entre a IoT e a computação em nuvem possibilita desenvolver, implantar, executar e gerenciar aplicativos por meio da internet (ZHOU *et al.*, 2013). Com a conexão entre essas duas tecnologias surge a nomenclatura *CloudThings* (ZHOU *et al.*, 2013). Nesse sentido, por meio do *CloudThings*, os modelos da computação em nuvem como *Infrastructure as a Service* (IaaS), *Platform as a Service* (PaaS) e *Software as a Service* (SaaS) podem estar integrados aos serviços de coleta de dados e ubíquos da IoT. A Figura 3.12 ilustra um exemplo de arquitetura em que são integrados a IoT e alguns dos modelos de serviços oferecidos pelas plataformas de computação em nuvem.

Figura 3.12 | Exemplo de conexão entre a IoT e os sistemas de computação em nuvem



Fonte: adaptada de Zhou *et al.* (2013).

No exemplo da Figura 3.12 foi utilizado o protocolo *Constrained Application Protocol* (CoAP) para realizar a comunicação entre os dispositivos IoT e as plataformas em nuvem, PaaS, SaaS e IaaS. O gateway é utilizado para garantir a troca eficiente de mensagens entre os dispositivos e as plataformas de computação em nuvem e é o responsável por traduzir as mensagens antes de enviá-las aos destinatários, que podem ser, por exemplo, os serviços oferecidos pela computação em nuvem. A comunicação na rede local de dispositivos é realizada via o protocolo 6LoWPAN, o qual apresenta uma similaridade e compatibilidade com os protocolos, HTTP e IPv6, comumente empregados na comunicação via internet.

Os serviços como armazenamento, visualização e compartilhamento, disponibilizados pela computação em nuvem, podem ser acessados pelos usuários através dos modelos SaaS, IaaS e PaaS.

Assimile

IaaS, PaaS e SaaS são modelos de serviços oferecidos pelas empresas de computação em nuvem. IaaS são os modelos de serviços de infraestrutura oferecidos pela computação em nuvem, por exemplo, ao invés de utilizar servidores locais, os provedores do serviço disponibilizam a

arquitetura em nuvem, para que o usuário utilize. No modelo de serviço PaaS possibilita criar, hospedar e gerir softwares nas plataformas de computação em nuvem. O modelo oferecido pelo SaaS consiste em utilizar softwares e ferramentas hospedadas nas plataformas de computação em nuvem. Um exemplo desse modelo são os serviços de e-mail e armazenamento (ERL; PUTTINI, 2013).

Nuvem de sensores

A união entre a IoT e a computação em nuvem gera avanços em ambas as áreas. Por exemplo, através da capacidade de processamento, virtualmente “ilimitado”, oferecido pela computação em nuvem, os dispositivos da IoT podem, a partir dos dados gerados, desenvolverem aplicações que utilizem algoritmos complexos de inteligência artificial. Esses algoritmos podem identificar padrões nessa massa de dados e geram informações valiosas sobre um determinado processo. Essas informações podem proporcionar uma melhor compreensão do processo e subsidiar decisões mais precisas sobre ações que visem melhorar toda a cadeia produtiva. Nesse sentido, a conexão entre IoT e a computação em nuvem pode contribuir com a melhoria de processos, produtos e serviços existentes através da união de sensores/atuadores e recursos compartilhados.

Quando vários sensores estão conectados de forma a construir uma infraestrutura abrangente capaz de proporcionar a interface entre o mundo físico e o virtual, através da internet, é denominado uma nuvem de sensores. A nuvem de sensores corresponde à virtualização dos sensores físicos. Essa virtualização é alcançada através do mapeamento de cada um dos sensores existentes na rede física.

A rede física corresponde aos sensores que estão presentes em diferentes ambientes físicos (reais) e realizam a comunicação através das Redes de Sensores Sem Fio (RSSF). Cada um dos sensores que compõe a nuvem possui um identificador que é utilizada no processo de virtualização. Por meio desse identificador é possível localizar o sensor, encontrar a região de monitoramento e quais os dados ele disponibiliza.

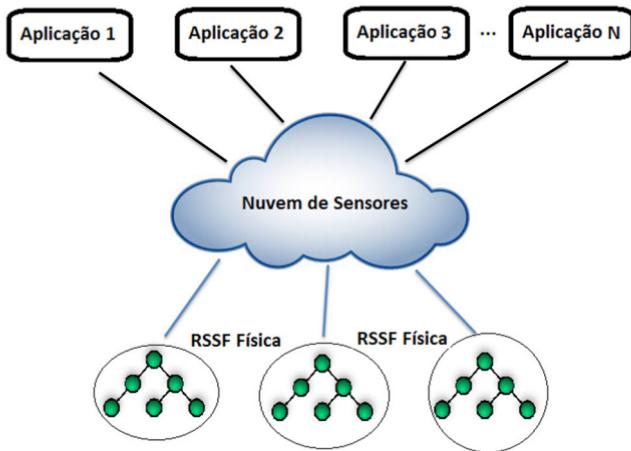
Esse tipo de arquitetura que envolve a virtualização possibilita o acesso remoto aos sensores, monitorar uma determinada área de interesse e realizar a interação com o ambiente através de requisições realizadas virtualmente.

A nuvem de sensores proporciona uma plataforma de visualização, armazenamento e gerenciamento de dados utilizando as tecnologias oferecidas pela computação em nuvem (ALAMRI *et al.*, 2013).

A nuvem de sensores amplia a possibilidade de integrar diferentes aplicações, uma vez que, existe a virtualização dos sensores, em outras palavras, o sensoriamento pode ser oferecido como um serviço ao usuário (DINH; KIM, 2016). Com isso, o usuário pode escolher o tipo de serviço disponibilizado pela rede de sensores, essa característica é similar aos serviços oferecidos pelos sistemas convencionais de computação em nuvem.

A Figura 3.13 apresenta um exemplo de arquitetura de nuvem de sensores. Nessa figura, estão presentes as aplicações, a nuvem de sensores e as RSSFs físicas. Para cada aplicação é definida uma identificação (ID), a área que deseja monitorar, um conjunto de sensores de interesse e o intervalo de aquisição das medidas.

Figura 3.13 | Modelo de arquitetura de nuvem de sensores



Fonte: adaptada de DINH e KIM (2016).

O usuário de uma nuvem de sensores pode realizar a consulta por um serviço da seguinte forma (LEMOS *et al.*, 2016): o usuário realiza uma consulta por um determinado parâmetro de alguma aplicação (temperatura, velocidade e umidade, etc.). Essa consulta é repassada para a nuvem de sensores através de um middleware. O middleware é o responsável por verificar os parâmetros do serviço que o usuário solicitou e, posteriormente, procurar por algum sensor que atenda a esses parâmetros e que esteja disponível para consulta. Após encontrar um sensor disponível, o usuário passa a ter acesso às informações solicitadas (LEMOS *et al.*, 2016). Assim, a nuvem de sensores garante acesso eficiente e rápido a várias aplicações através de um método elaborado de consultas que é transparente ao usuário.

Exemplos de redes de sensores

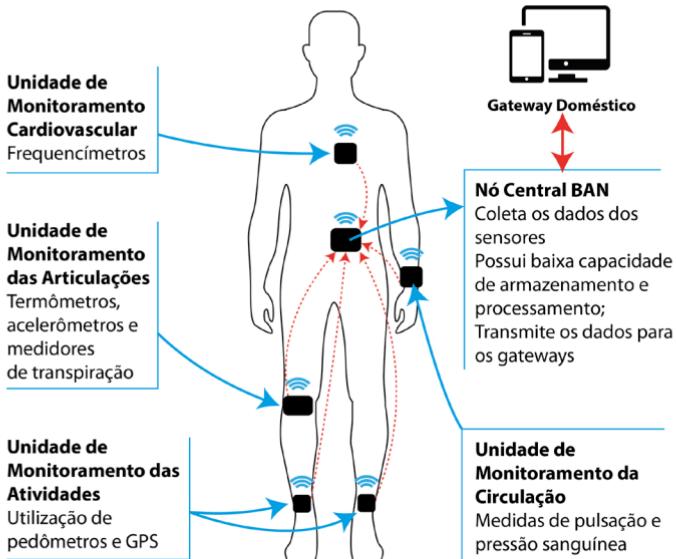
Como você deve ter observado, com a união dos dispositivos da IoT com as plataformas de computação em nuvem, temos um cenário ideal para o desenvolvimento de aplicações que possam explorar todo o potencial gerado pela união dessas duas tecnologias. Com a utilização dessas tecnologias, em especial através das redes de sensores sem fio, é possível monitorar eventos ou desastres ambientais.

Uma aplicação em que podemos evidenciar a importância das redes de sensores é a construção de edifícios e casas inteligentes (BYUN *et al.*, 2012), (SURYADEVARA *et al.*, 2014) e (GHAYVAT *et al.*, 2015). A construção de edifícios e residências inteligentes é realizada por meio do monitoramento constante de vários indicadores. Por exemplo, com a conexão de vários sensores pode-se monitorar, em tempo real, o consumo de energia de toda a residência. Logo, o mapeamento do consumo possibilita encontrar padrões de utilização de energia. Com isso, estabelecer os horários de pico de consumo. Com esses dados, pode-se realizar um planejamento eficiente para a redução dos gastos com energia (SURYADEVARA *et al.*, 2014).

Outra aplicação pode estar na melhoria da qualidade de vida dos idosos (MAJUMDER *et al.*, 2017) e na telemedicina (PERUMAL *et al.*, 2012). Para a melhoria da qualidade de vida dos idosos, tem-se um projeto chamado *Wireless Body Area Network* (WBAN). Nessa rede, existem subunidades, constituídas por sensores responsáveis por monitorar diferentes áreas do corpo como coração, membros e atividade motora. Ao coletar esses dados eles são enviados para a unidade central. A unidade central é utilizada como um gateway que permite a conexão da WBAN com outras redes da residência e a internet.

A Figura 3.14 apresenta um esquema do WBAN. Para a comunicação entre os dispositivos vestíveis (unidades), são utilizados os protocolos baseados no IEEE 802.15.4/Zigbee (DARWISH; HASSANIEN, 2011).

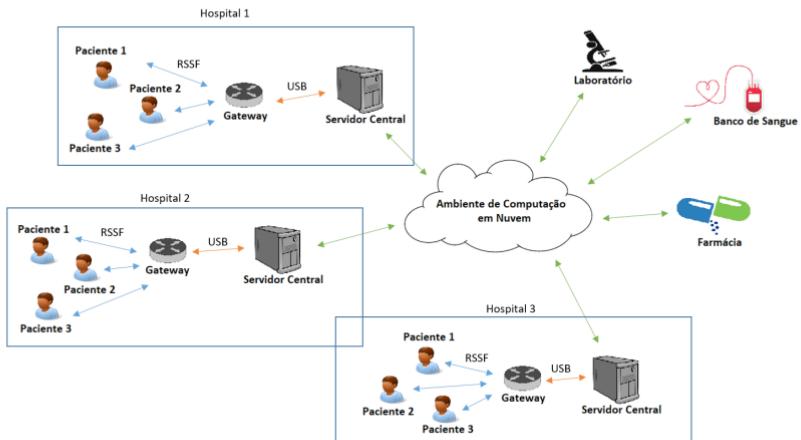
Figura 3.14 | Exemplo de aplicação da rede de sensores no monitoramento de idosos



Fonte: adaptada de Majumder *et al.* (2017).

A telemedicina consiste em utilizar a tecnologia para auxiliar nos cuidados com a saúde em locais onde a distância entre o paciente e os centros de saúde é um fator crítico (WEN, 2008). A Figura 3.15 mostra a arquitetura utilizada nessa aplicação, em que é realizado o monitoramento de alguns pacientes em diferentes hospitais. Esse monitoramento é realizado por meio de redes de sensores sem fio responsáveis por medir os parâmetros fisiológicos dos pacientes como eletrocardiograma (ECG), batimentos cardíacos e temperatura corporal. Após coletar esses dados eles são transmitidos para um gateway e, através da conexão *Universal Serial Bus* (USB), para um servidor central existente em cada um dos hospitais. Os dados coletados e presentes no servidor são analisados e compartilhados por meio da internet. Essas informações podem ser compartilhadas entre diferentes integrantes dos sistemas de saúde como laboratórios, farmácias e bancos de sangue (Perumal *et al.*, 2012).

Figura 3.15 | Exemplo de aplicação das redes de sensores na telemedicina



Fonte: adaptada de Perumal *et al.* (2012).

A utilização das redes de sensores pervasivas em diferentes campos da medicina tem por objetivo fornecer informações em tempo real e informar sobre eventos que possam indicar uma possível situação anormal dos pacientes (ALEMDAR; ERSOY, 2010). O monitoramento constante possibilita gerar maior independência para idosos e doentes crônicos.

Reflita

Na aplicação sobre a telemedicina de pacientes, foi mostrado que a comunicação do gateway com a central é realizada através da conexão USB. Você poderia indicar algumas sugestões de melhoria para essa aplicação?

A agricultura de precisão também utiliza os benefícios gerados pelas redes de sensores para melhorar os processos e a qualidade dos produtos gerados (ABBASI *et al.*, 2014) e (TONGKE, 2013). Para a melhoria da qualidade dos produtos é necessário o monitoramento dos constituintes do solo, da qualidade da água e das variáveis climáticas. A partir desse monitoramento é possível utilizar sistemas de controle que possibilitem controlar, por exemplo, os processos de irrigação, o crescimento e combate a pragas e a variação de nutrientes presentes no solo (ABBASI *et al.*, 2014).

Consolidação de dados coletados via IoT

Para utilizar todo esse potencial que IoT proporciona, é necessário que os dados gerados possam ser integrados e analisados de maneira eficiente. A grande aliada da IoT na consolidação e análise dos dados de maneira eficiente é a computação em nuvem (SOLDATOS, 2016). Através da capacidade, escalabilidade e desempenho dos serviços oferecidos pela computação em nuvem é que os dados obtidos podem ser integrados e analisados de forma a gerarem informações que agreguem valor ao produto e à própria empresa.

Pesquise mais

O artigo referenciado a seguir, descreve detalhadamente a construção de uma casa inteligente. Nesse artigo, também são descritas situações que ocorrem no dia a dia e como as redes de sensores, a computação em nuvem e a IoT propiciam maior comodidade e eficiência na realização das tarefas diárias.

ZHOU, J. et al. Cloudthings: *A common architecture for integrating the internet of things with cloud computing*. In: **Proceedings of the 2013 IEEE 17th international conference on computer supported cooperative work in design (CSCWD)**. IEEE, 2013. p. 651-657.

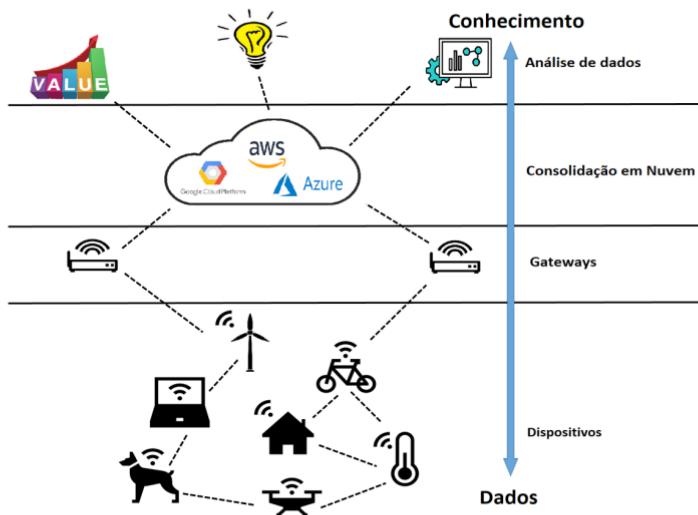
Os dados gerados pelos diferentes sensores são integrados e analisados e podem ser armazenados e consolidados utilizando as plataformas em nuvem. A partir dessa consolidação dos dados e da aplicação de técnicas, como a mineração de dados, é que o sensoriamento pervasivo realizado pela IoT pode gerar conhecimento, tornando os processos mais eficientes.

A Figura 3.16 apresenta um exemplo de como os dados coletados podem ser consolidados por meio das plataformas em nuvem. Existem várias ferramentas comerciais que podem ser utilizadas como sistemas de consolidação desses dados. Algumas das ferramentas que podem ser citadas são: o Microsoft Azure IoT, o Google Cloud IoT e o Amazon Web Services (AWS) IoT. Tais ferramentas proporcionam um rápido e eficiente ambiente de integração de dados de diferentes fontes e permitem a aplicação de algoritmos para a construção de conhecimento a partir desses dados.

A comunicação entre os dispositivos da rede local ou entre esses dispositivos e os gateways pode ocorrer por meio de diferentes protocolos, como 6LoWPAN, MQTT, Bluetooth e ZigBee. Nesse sentido, os dados são enviados para os gateways e repassados para as plataformas em nuvem.

Utilizando as ferramentas de consolidação, os dados de diferentes fontes podem ser integrados na nuvem. Diante disso, é possível aplicar algoritmos que permitem encontrar padrões e extrair conhecimentos.

Figura 3.16 | Exemplo de conexão dos dispositivos para consolidação dos dados via IoT



Fonte: elaborada pelo autor.

Para alcançar uma integração eficiente dos dados é necessário conhecer as particularidades das aplicações IoT. Quando são comparadas a forma como são gerados e o conteúdo dos dados originados das aplicações IoT com o Big Data convencional constata-se a existência de algumas diferenças (SOLDATOS, 2016). Por exemplo, o Big Data convencional, apresenta um volume de dados proveniente de grandes depósitos de dados (warehouses) e numerosas fontes. Enquanto que no Big Data IoT, possui um volume de dados provenientes de sensores e dispositivos conectados à internet.

Como pode ser visto pelos exemplos de aplicações listados anteriormente, encontramos os sensores e dispositivos IoT realizando diferentes funções em cenários variados. Essa diversidade de elementos e heterogeneidade de aplicações é uma característica da Internet das Coisas, que proporciona a aquisição de dados com formatos ($^{\circ}\text{C}$, Volts, Amperes, Watts etc.), sentido (temperaturas, umidade do ar/solo em diferentes locais) e velocidades de aquisição (frequências) diferentes.

O tratamento dessa inerente heterogeneidade dos dados é um dos desafios encontrados na consolidação e análise da informação para a IoT

(SOLDATOS, 2016). Como os dispositivos da IoT possuem uma diversidade de fabricantes e que muitas vezes proporcionam diferentes incertezas nas medidas geradas pelos sensores, tratar e avaliar esses dados torna-se uma tarefa complexa para os sistemas convencionais. Assim, é necessário adaptar as estratégias existentes ou desenvolver novas ferramentas que possibilitem tratar de forma eficiente os dados originados de aplicações IoT.

Exemplificando

Um exemplo de sucesso de integração entre os diferentes tipos de Big Data é o realizado por empresas de logística e transporte, que utilizam sensores espalhados em cada um dos veículos, mapas de rotas e algoritmos de aprendizado de máquina para encontrar as melhores rotas de entregas e reduzir os custos de transporte (JAIN *et al.*, 2017).

Segundo (SOLDATOS, 2016), para garantir a consolidação e análise eficiente dos dados da IoT podem ser aplicadas técnicas estatísticas, de aprendizado de máquina, mineração de dados e gerenciamento de fluxo e banco de dados. A união dessas técnicas é que possibilita tratar de forma eficiente as características do Big Data IoT. Portanto, é possível realizar a consolidação e análise dos dados de forma precisa, o que permite obter informações e conhecimento relevante sobre a aplicação. Assim, a IoT pode ser utilizada como meio de gerar melhores produtos e otimizar processos.

Chegamos ao final de mais uma seção. Nessa caminhada aprendemos um pouco mais sobre como a união entre a IoT e a computação em nuvem proporciona o desenvolvimento de aplicações mais robustas e confiáveis. Espero que você tenha compreendido o potencial transformador promovido pela conexão dessas tecnologias.

Sem medo de errar

Agora você já conhece toda a importância que a computação em nuvem tem para o desenvolvimento da IoT. Você deve ter percebido que a união dessas duas tecnologias gera grandes oportunidades de melhorias nos produtos, processos e serviços existentes.

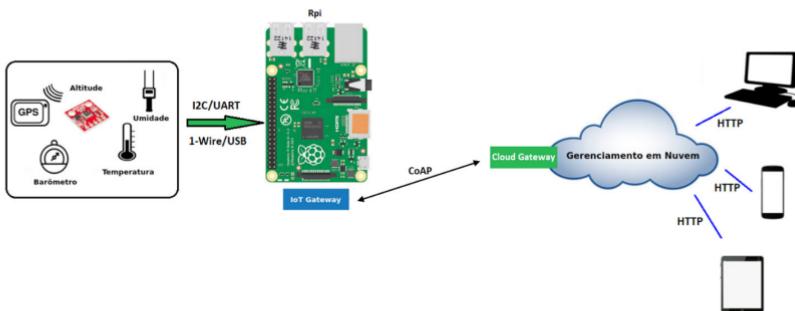
Lembre-se, você é o responsável por desenvolver um instrumento inteligente capaz de auxiliar na melhoria da produção de uma grande fazenda de produção de soja. Esse instrumento inteligente deve ser capaz de monitorar as condições climáticas, integrar os dados coletados com a internet e

possibilitar que, a partir da análise desses dados, sejam tomadas decisões que melhorem todo o processo produtivo da empresa.

Você está na segunda etapa de desenvolvimento desse instrumento inteligente. Agora você deve completar o relatório que iniciou na primeira etapa. Nesta segunda etapa, esse relatório deve conter a indicação de como os sensores de temperatura, pressão atmosférica, velocidade do vento, radiação solar e umidade do ar devem ser conectados à internet. Além disso, esse relatório precisa conter as vantagens que a união entre os dados gerados pelo instrumento e a computação em nuvem oferece. Essas informações são necessárias, pois os clientes precisam estar convencidos de que a união entre o instrumento de medição e a internet pode viabilizar a melhoria de toda a produção.

Uma possível indicação de união entre os dados coletados pelo instrumento de medição e a computação em nuvem está representada na Figura 3.17, que mostra como os sensores são conectados ao Rpi por meio de vários protocolos diferentes. Isso ocorre, pois cada um dos sensores apresenta uma forma de comunicação (saída de dados) diferente. Após a integração desses sensores com o Rpi, existe a possibilidade de realizar a comunicação através da internet. Para isso, é utilizado o protocolo CoAP, pois ele apresenta características similares ao HTTP e foi desenvolvido para dispositivos com fonte limitada de recursos. Para realizar a tradução das mensagens trocadas entre o instrumento inteligente e a nuvem é utilizado um gateway. Assim, por meio dessa figura é possível identificar que a união entre a IoT e a computação e nuvem pode ocorrer através de diferentes protocolos. Nesse caso, é necessário que as duas redes possam “conversar” de maneira eficiente, ou seja, por meio dos protocolos e gateways, as mensagens trocadas devem ser compreendidas entre ambas as tecnologias.

Figura 3.17 | Sugestão de conexão entre o instrumento inteligente e a internet



Fonte: elaborada pelo autor.

Com a utilização dessas duas tecnologias toda a complexidade do sistema será transparente ao usuário. Assim, ele receberá apenas as informações solicitadas, sem a necessidade de conhecer todo o processo de construção. O usuário também pode acessar a informação de maneira remota através de diferentes plataformas como smartphones, tablets ou computador. O processamento e armazenamento da informação também devem ocorrer de forma eficiente e distribuída. O usuário pode coletar dados de vários sensores que estejam localizados em uma mesma região e, por exemplo, através de uma análise estatística mais detalhada, encontrar valores mais precisos da medição. Com isso, os algoritmos de análise de dados, que devem ser implementados nas plataformas em nuvem, podem gerar informações ou alertas precisos sobre as condições da plantação. De posse desses dados podem ser produzidas informações como previsão de colheita, saúde das plantas, melhores períodos para a rotatividade do solo ou cultura, ajuste da quantidade de fertilizantes e pesticidas e criação de um calendário de irrigação tudo isso de maneira automática. Assim, a gerência dessa fazenda pode tomar decisões mais assertivas sobre o planejamento de toda a produção. Esse maior conhecimento sobre a cultura aumenta os lucros da empresa, melhora a produtividade e qualidade das mercadorias oferecidas.

Avançando na prática

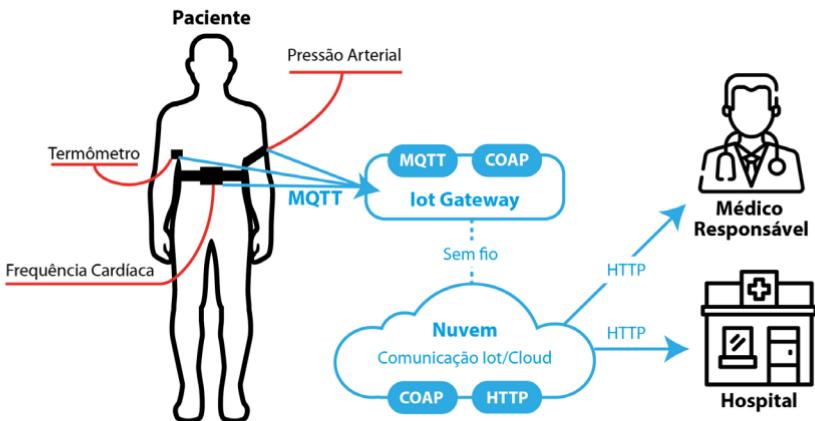
Acompanhamento remoto de pacientes

Durante o período em que estava trabalhando no desenvolvimento do instrumento inteligente para o monitoramento da fazenda de soja, o chefe da empresa em que trabalha, pediu para que você ajudasse a escrever o relatório sobre como devem ser utilizados os sensores/atuadores de outra aplicação que a empresa está desenvolvendo. Essa aplicação consiste no monitoramento remoto de pacientes. Na primeira parte do relatório que você recebeu já contém os sensores a serem utilizados (pressão arterial, batimentos cardíacos e temperatura corporal). Assim, a sua tarefa é completar esse relatório indicando como esses sensores devem ser integrados com a computação em nuvem e mostrar as possibilidades que essa união de tecnologias gera para a melhoria do acompanhamento de pacientes.

Resolução da situação-problema

A Figura 3.18 mostra um modelo de conexão entre os dispositivos corporais (*wearable sensors*) e as plataformas em nuvem.

Figura 3.18 | Exemplo de solução para o acompanhamento remoto de pacientes



Fonte: elaborada pelo autor.

Para a aplicação no monitoramento de pacientes, a união entre a IoT e a computação em nuvem proporciona grandes melhorias no bem-estar e na qualidade de vida dos pacientes assistidos. Através dessa conexão é possível que vários diferentes indicadores da saúde dos pacientes sejam coletados e consolidados através das plataformas em nuvem. Alguns sensores como medidor de frequência cardíaca, pressão arterial e termômetro podem ser empregados para monitorar a saúde do paciente. Esses sensores podem utilizar diferentes protocolos para realizar a comunicação como ZigBee ou MQTT. Após esses dados serem coletados é necessário enviá-los para plataformas em nuvem. Na comunicação com a internet é utilizado o CoAP. A comunicação entre a nuvem e a rede local é realizada via um gateway. Esse gateway pode ser, por exemplo, um Raspberry Pi. Ele possibilita a junção dos dados dos diferentes sensores “vestíveis” e disponibiliza-os através da internet.

A Figura 3.17 apresenta as transcrições realizadas pelo gateway. Ao serem enviados para a nuvem, esses dados podem ser acessados por diferentes serviços médicos através de requisições HTTP, por exemplo. Assim, a visualização dos dados pode ser realizada de maneira remota pelos centros médicos, o que propicia uma maior independência aos pacientes.

A partir dos dados consolidados pode-se realizar uma análise mais criteriosa com o auxílio de algoritmos de aprendizado de máquina. Com isso, caso seja identificado o comportamento anormal nos dados, podem ser criados alertas. Esse alerta pode ser enviado, diretamente, ao centro médico para que as medidas necessárias sejam tomadas pelos médicos responsáveis. Esses são exemplos de benefícios gerados pela integração dos dados através das plataformas em nuvem. Utilize o conhecimento adquirido e sua imaginação para pensar em outros benefícios aos pacientes monitorados.

Faça valer a pena

1. Os dispositivos da Internet das Coisas são, normalmente, compostos por sensores/atuadores com reduzida disponibilidade de recursos. Quando esses dispositivos utilizam as características oferecidas pela computação em nuvem, eles podem obter um incremento de desempenho através da virtualização de serviços.

Assinale a alternativa que contém as características da computação em nuvem que propiciam o incremento de desempenho para a IoT.

- a. Visualização remota de dados e melhorias de hardware.
- b. Aumento do número de dispositivos e maiores taxas de transferência.
- c. Compartilhamento de banda de transferência de dados e visualização de processos.
- d. Compartilhamento de recursos e serviços.
- e. Aumento no fornecimento de energia e visualização remota.

2. A integração entre a IoT e as plataformas de computação em nuvem gera grandes oportunidades de desenvolvimento para ambas as tecnologias.

Sobre as oportunidades geradas pela união dessas tecnologias, análise as afirmativas a seguir.

- I. Virtualização de processamento e armazenamento.
- II. Possibilidade de visualização remota dos dados.
- III. Possibilidade de maior interação entre o ambiente físico e virtual.
- IV. Encapsulamento da complexidade para acesso aos dados gerados pela IoT.

A respeito das oportunidades geradas pela união da IoT com as plataformas em nuvem, é correto o que se afirma em:

- a. I e II, apenas.
- b. I e III, apenas.
- c. I, II e IV, apenas.
- d. I, apenas.
- e. I, II, III, IV.

3. A integração entre a IoT e os sistemas de computação em nuvem geram inúmeras possibilidades para ambas as tecnologias. Entretanto, para uma integração eficiente entre elas é necessário que algumas características particulares dos dados gerados pela IoT sejam observadas.

Assinale a alternativa correta que explica as dificuldades encontradas para a integração da computação em nuvem e a IoT.

- a. Os dados gerados pelos sensores da IoT apresentam baixa velocidade de aquisição, são compostos por dispositivos com reduzida fonte de energia e alta capacidade de processamento o que dificulta a geração de dados precisos sobre os processos.
- b. Os dados gerados pelos sensores da IoT apresentam alta velocidade de aquisição, provém de diferentes fabricantes e, normalmente, possuem baixa capacidade de processamento o que dificulta a utilização de estratégias convencionais.
- c. Os processamentos realizados pelas plataformas em nuvem não são capazes de tratar os dados da IoT, pois o armazenamento é limitado, apesar do processamento robusto, os sistemas em nuvem não conseguem lidar com o fluxo de dados gerados pela IoT.
- d. O processamento local ainda continua sendo a melhor alternativa para a IoT. Isso ocorre, pois, os serviços oferecidos pelas plataformas em nuvem não são eficazes no tratamento do fluxo de dados da IoT.
- e. Os dados gerados pela IoT não são tratados de maneira eficiente pelas plataformas em nuvem, pois o processamento e armazenamento oferecidos apresentam-se como uma barreira para o tratamento eficiente do fluxo de dados gerados pela IoT.

Seção 3

Manipulação e análise de dados na IoT

Diálogo aberto

Você já parou para pensar sobre as oportunidades e facilidades geradas pela internet? A internet encurtou distâncias, facilitou o acesso à informação, possibilitou conhecer o mundo por meio de um computador e proporcionou o desenvolvimento de novas tecnologias, como a Internet das Coisas. Há um tempo não muito distante, para realizar uma pesquisa escolar ou apenas retirar uma pequena dúvida sobre um fato histórico, era necessário recorrer a uma enormidade de livros e passar minutos ou horas pesquisando em várias páginas de enciclopédias. Hoje, com alguns toques na tela do smartphone, é possível acessar, praticamente, todo tipo de informação desejada em apenas alguns segundos. Com o advento da IoT, as facilidades geradas pela internet são ainda maiores.

A IoT expande as possibilidades geradas pela internet, pois agora não existe apenas a união virtual entre várias páginas de busca ou enciclopédias, temos a oportunidade de conectar os objetos reais do nosso cotidiano por meio dessa rede de tal forma que ações realizadas virtualmente provoquem alterações no mundo real. Por exemplo, o seu próprio automóvel pode identificar que está próximo a um período de manutenção preventiva e realizar, automaticamente, o agendamento desse serviço. Essa facilidade é possível graças à união entre a IoT e as plataformas em nuvem.

As plataformas em nuvem proporcionam inúmeras possibilidades para aplicações IoT. Com a utilização da internet é possível integrar os sensores, manipular, visualizar e analisar os dados remotamente. Você já imaginou o quanto essas possibilidades impactam na forma como o monitoramento e interações com o ambiente são realizadas? Com a integração dos dados em nuvem podemos processar os dados e gerar insumos para a tomada de decisão sem a necessidade de coletar, localmente, toda essa informação. Isso agiliza os processos e garante o monitoramento remoto do ambiente.

Lembre-se que você está trabalhando na construção de um sistema inteligente conectado à internet que seja capaz de monitorar o microclima em uma plantação de soja. Esse sistema também deve possibilitar que os dados coletados estejam disponíveis para visualização e análise em plataformas em nuvem. Para isso, você deve utilizar o Rpi para conectar sensores/atuadores, enviar os dados para a nuvem e processá-los a fim de obter informações que auxiliem no melhor aproveitamento da lavoura.

Você chegou à última etapa de desenvolvimento dessa aplicação. Os diretores e clientes esperam que você complete o relatório adicionando um esquema que contenha a conexão de todos os elementos necessários para a construção do instrumento inteligente. Esse esquema deve conter desde os sensores até algumas ferramentas utilizadas para a análise dos dados. Dentre os elementos intermediários desse esquema deve estar presente a conexão dos sensores e construção do Big Data. A fim de auxiliá-lo nessa tarefa, no decorrer desta seção, você vai aprender como utilizar o Rpi para a integração de sensores e acesso aos serviços em nuvem. Além disso, você vai aprender sobre o Big Data da IoT e conhecer alguns modelos que permitem manipular e analisar essa massa de dados.

Agora é o momento de você mostrar aos diretores e clientes as habilidades que adquiriu ao longo dessa jornada através da IoT. Chegou a hora de desenvolver um esquema para o protótipo que aplica vários conceitos aprendidos durante essa caminhada. Este é o momento de você se destacar no mercado. Está preparado? Então, vamos lá.

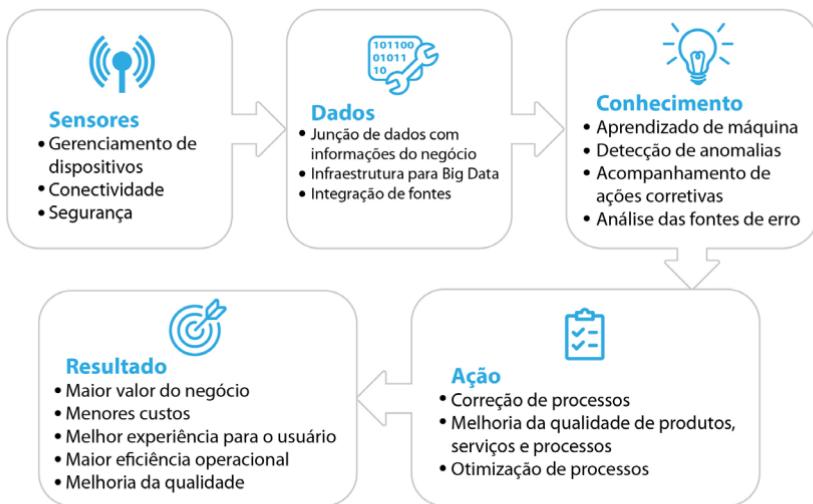
Não pode faltar

A internet pode oferecer aos dispositivos da IoT muito mais que um meio de comunicação. Através da computação em nuvem é possível oferecer vários serviços sob demanda aos dispositivos da IoT e, assim, proporcionar maior capacidade de processamento e armazenamento. Essa união viabiliza o desenvolvimento de inúmeras aplicações. Por meio da integração entre sensores e a computação em nuvem torna-se viável empregar tecnologias como o aprendizado de máquina às aplicações da IoT. Essas tecnologias podem ser empregadas através da virtualização de sensores proporcionada pelos serviços da computação em nuvem (MANEA; POPA, 2016). Além disso, a virtualização de sensores facilita o processo de conexão de diferentes usuários a uma mesma aplicação e auxilia no desenvolvimento de aplicações em diversos ambientes (HANG *et al.*, 2018).

Por meio da virtualização dos sensores, a realização da análise de dados pode ser executada de forma mais precisa e eficiente. Atualmente, encontrar ou produzir dados não é um desafio como há anos. Com bilhões de dispositivos conectados à internet, gerando trilhões de bytes em milhares de aplicações diferentes o desafio enfrentado pelos cientistas, engenheiros e empresas passa a ser o de garantir que essa grande variedade de dados possa ser transformada em informações que gerem melhorias nos produtos e processos produtivos. Para assegurar que informações relevantes e de qualidade sejam extraídas dessa massa de dados é necessário seguir alguns passos

e realizar uma análise criteriosa sobre esses dados (GROUP, 2017). A Figura 3.19 apresenta uma visão geral sobre os passos a serem seguidos do momento da coleta dos dados até alcançar informações que gerem valor aos negócios.

Figura 3.19 | Passos para a transformação dos dados em valor de negócio



Fonte: adaptada de Group (2017).

Quando os dados da IoT são, corretamente, consolidados a análise de dados pode, efetivamente, auxiliar na otimização de processos. A análise de dados corresponde ao processo utilizado para examinar um conjunto de dados a fim de encontrar padrões, estatísticas, preferência de clientes, correlações e tendências de mercado que possibilitem extrair informações relevantes e conhecimento a partir desses dados (MARJANI *et al.*, 2017). A manipulação de dados necessita de tecnologias que possam transformar uma grande variedade de dados estruturados e não estruturados em modelos capazes de serem interpretados pelas ferramentas de análise. Essas ferramentas correspondem a algoritmos capazes de reconhecer padrões, tendências e correlações nessa massa de dados. Além disso, essas ferramentas devem permitir a criação de gráficos e tabelas que auxiliem na visualização e apresentação desses dados em formas mais amigáveis para a análise. Assim, é possível que o conjunto de dados, anteriormente, não estruturados possa propiciar a realização da tomada de decisão (MARJANI *et al.*, 2017).

Para cada tipo de aplicação de IoT é necessário que seja executado um tipo específico de análise, pois a natureza dos dados coletados influencia nos resultados obtidos. A análise dos dados pode ser classificada em análise de tempo real, análise off-line, análise em nível de memória, análise de

inteligência de negócio (do inglês *Business Intelligence* - BI) e análise massiva (MARJANI *et al.*, 2017). O Quadro 3.4 apresenta uma comparação entre os diferentes sistemas de análise de dados.

Quadro 3.4 | Comparaçao entre os diferentes tipos de análise

Tipo de Análise	Uso Específico	Arquiteturas e Ferramentas Existentes	Vantagens/ Categorias
Tempo Real	Analizar um grande número de dados gerados por sensores.	Greenplum HANA	Processamento paralelo. Utilização de banco de dados para processos de clusterização.
Off-Line	Utilizado em aplicações que não possuem grande compromisso com o tempo de resposta.	Scribe Kafka Timetunnel Chukwa	Aquisição eficiente de dados. Reducir o custo de formatação e conversão dos dados.
Nível de Memória	Utilizado quando o volume de dados é menor que a capacidade de memória disponível no cluster.	MongoDB	Tempo real.
Nível de Inteligência de Negócio	Utilizado quando a escala dos dados ultrapassou o nível de memória.	Planos de análise de dados	Utilização off-line e online.
Nível Massivo	Utilizado quando a quantidade de dados ultrapassa completamente a capacidade dos produtos de inteligência de negócios e os bancos de dados tradicionais.	MapReduce	Normalmente, off-line.

Fonte: adaptado de Marjani *et al.* (2017).

A integração entre os dispositivos da IoT e a internet é fundamental para a geração de conhecimento. A integração dos dispositivos possibilita a consolidação, armazenamento e análise dos dados gerados pelas aplicações IoT.

	<p>Você pode acessar um tutorial de desenvolvimento de uma aplicação da Internet das Coisas que realiza a integração dos dados e consolidação através da internet. Esse tutorial pode ser acessado através do QR Code ou link: http://cm-cls-content.s3.amazonaws.com/ebook/embed/qr-code/2019-2/internet%20das%20coisas/u3/s3/QRCODE_IoT_3.3_1pv.pdf</p>
-------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Segundo Marjani *et al.* (2017), o gerenciamento de dados para a IoT pode ser dividido em 3 etapas.

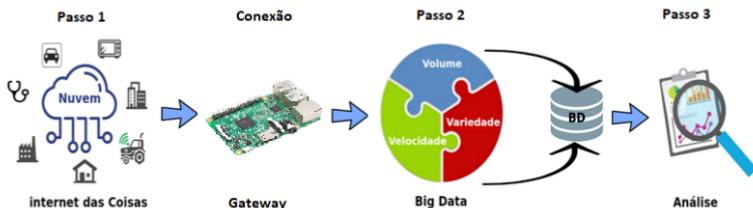
- O **primeiro passo** consiste em promover o gerenciamento das fontes de dados, ou seja, definir como os dispositivos devem interagir uns com os outros.
- No **segundo passo**, esses dados são chamados de “Big Data” devido ao volume, velocidade e variedade. Assim, deve existir um armazenamento eficiente desses dados. Para possibilitar que os dados gerados pelos sensores sejam enviados e armazenados pelas plataformas em nuvem é necessária a utilização de um gateway que permita conectar os sensores da IoT a essas plataformas.

O processo de tradução entre os diferentes protocolos, realizado pelo gateway, não é considerado um passo, pois os dados que entram no gateway não sofrem alteração. Existe apenas uma tradução de mensagens para que os dados sejam corretamente interpretados nas plataformas em nuvem. Após passarem pelo gateway, ocorre o último passo desse gerenciamento.

- O **último passo** consiste em aplicar as ferramentas de análise nesse conjunto de dados armazenados para extrair as informações relevantes que possam garantir a tomada correta de decisão.

A Figura 3.20 apresenta uma representação gráfica para o processo relatado.

Figura 3.20 | Passos para o gerenciamento eficiente dos dados da IoT



Fonte: adaptada de Marjani *et al.* (2017).

Uma questão importante para viabilizar a utilização dos sistemas de computação em nuvem e análise dos dados é a escalabilidade. Isso ocorre, pois a IoT pode ser composta por dezenas ou milhares de dispositivos (TRUONG *et al.*, 2016). A escalabilidade possibilita construir uma rede com apenas alguns dispositivos conectados e, quando necessário, aumentar a quantidade de dispositivos sem ter que refazer toda a estrutura existente. Quando a análise de dados utiliza a elasticidade dos serviços sob demanda (armazenamento e processamento) oferecidos pelas plataformas em nuvem, é possível que os algoritmos utilizados para a análise de dados possam operar sobre uma base de dados com tamanho variável. Assim, dependendo da aplicação e das necessidades dos usuários a análise do Big Data pode ser

realizada de maneira customizada, o que garante a utilização eficiente dos serviços oferecidos.

Assimile

As características que definem um conjunto de dados como Big Data são chamadas de 3 V's. Esses 3 V's correspondem ao volume, variedade e velocidade (SU *et al.*, 2018). O volume corresponde à grande quantidade de dados produzidos. A variedade ocorre devido à existência de diferentes fontes geradoras de dados, que podem vir, por exemplo, de sensores, da gerência, de previsões realizadas ou de outras diferentes fontes. A alta velocidade com que esses dados são produzidos também caracteriza o Big Data. Assim, essas características definem um conjunto especial de dados que devem ser armazenados e tratados por meio de técnicas especiais.

Segundo Minteer (2017), o volume de dados gerados pelas aplicações IoT cresce rapidamente e os gastos computacionais para realizar a análise desses dados é muito intenso. Portanto, para assegurar uma análise correta dos dados e que as informações obtidas a partir dessa análise possam ser utilizadas para gerar valor ao negócio é necessário garantir que essa análise seja escalável, ou seja, possa utilizar uma base de dados gerada por 100 ou 1 milhão de dispositivos. Uma análise escalável reduz a complexidade, melhora a manutenibilidade do sistema, reduz custos e, assim, possibilita a realização de análises mais complexas. Por meio de análises mais complexas e criteriosas existe uma maior probabilidade de encontrar padrões e aumentar o valor do negócio. Essas ações geram um ciclo que promove uma melhoria constante das organizações.

O Big Data gerado pela IoT pode ser convertido em conhecimento através da mineração de dados (do inglês Data Mining). Devido às características especiais dos dados gerados pela IoT (heterogeneidade, velocidade de aquisição e veracidade), os algoritmos tradicionais para mineração de dados não podem ser diretamente empregados. Assim, coletar, analisar e gerenciar os dados da IoT envolve a atualização e adequação dos algoritmos existentes (GUPTA; GUPTA, 2017).

Reflita

Devido à grande quantidade de dispositivos, oriundos de diferentes fabricantes e desempenhando funções variadas, a IoT apresenta-se como uma das principais tecnologias responsáveis pela criação do Big Data. Sendo assim, a IoT impõe desafios a serem enfrentados pelos algoritmos e sistemas

tradicionais nas tarefas de coletar, armazenar e analisar os dados. Por exemplo, como identificar se os dados coletados por sensores apresentam anomalias. Você saberia elencar outros desses desafios e propor soluções?

Mineração de dados é uma tecnologia utilizada para identificar tendências ou padrões desconhecidos em um conjunto de dados com o objetivo de melhorar a performance das organizações (GUPTA; GUPTA, 2017). É através da mineração de dados que todo o conjunto de dados coletados pelos dispositivos IoT em diferentes aplicações são estruturados e as informações relevantes, que possam melhorar a eficiência dos processos, são extraídas. Com a utilização correta dos algoritmos de mineração de dados sobre os dispositivos da IoT, não apenas as indústrias são beneficiadas. Devido à característica ubíqua da IoT, toda a sociedade pode aproveitar essas informações para melhorar o ambiente em que vive. As informações geradas através dos objetos inteligentes presentes nos sistemas de transporte, nas unidades de saúde, no monitoramento da segurança patrimonial e de pessoas, além de economia de energia, representam uma grande fonte de informação para tornar as cidades um local mais agradável, dinâmico e que preserva o meio ambiente e as relações entre pessoas. Portanto, garantir que informações relevantes sejam obtidas através das aplicações de IoT representa um dos principais objetivos da utilização da mineração de dados na IoT (GUPTA; GUPTA, 2017).

Reflita

Devido à grande quantidade de dispositivos, oriundos de diferentes fabricantes e desempenhando funções variadas, a IoT apresenta-se como uma das principais tecnologias responsáveis pela criação do Big Data. Sendo assim, a IoT impõe desafios a serem enfrentados pelos algoritmos e sistemas tradicionais nas tarefas de coletar, armazenar e analisar os dados. Por exemplo, como identificar se os dados coletados por sensores apresentam anomalias. Você saberia elencar outros desses desafios e propor soluções?

A análise de dados através da internet é fundamental para o entendimento de processos e construção do conhecimento. Além disso, algoritmos de mineração de dados e aprendizado de máquina podem ser empregados para identificar anomalias nos dados gerados pelos sensores.



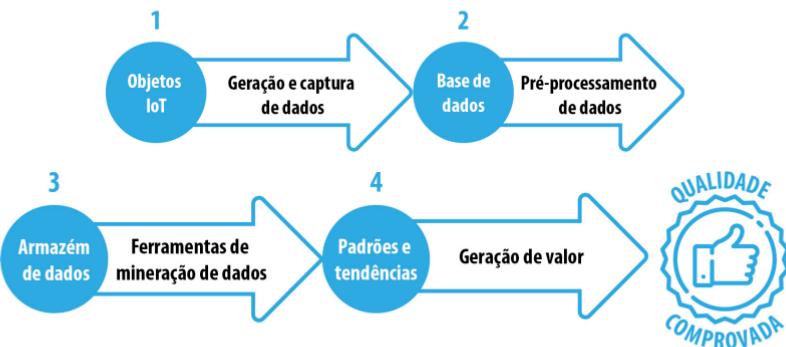
Através do QR Code ou link a seguir, você pode acessar uma aplicação que demonstra a utilização do algoritmo K-mens na análise de dados e detecção de anomalias em dados de temperatura gerados por sensores.
http://cm-kls-content.s3.amazonaws.com/ebook/embed/qr-code/2019-2/internet%20das%20coisas/u3/s3/QRCODE_IoT_3.3_1pv.pdf

A Figura 3.21 mostra um modelo para a análise dos dados da IoT. Como pode ser visto, para que os algoritmos de mineração sejam capazes de extrair informações importantes que possam melhorar a eficiência de processos existentes, são necessários alguns passos essenciais.

- O primeiro passo corresponde à **captura dos dados** através dos sensores.
- No segundo passo, esses dados são consolidados e deve ocorrer a limpeza dos dados. Essa limpeza, ou pré-processamento, consiste em remover dados sem significado (dados errados ou nulos).
- O terceiro passo corresponde em agrupar os dados da IoT com outras fontes de dados como redes sociais ou gerenciais.

Após essa fase podem ser aplicadas as ferramentas de mineração de dados a fim de identificar padrões e tendências. A partir desses padrões é possível gerar conhecimento sobre o processo. A partir da aplicação eficiente desses algoritmos é possível gerar valor de negócios para as empresas e produzir melhorias nos serviços prestados. Com isso, toda a sociedade será beneficiada pela utilização eficiente da mineração de dados sobre o Big Data gerado pela IoT.

Figura 3.21 | Exemplo de framework para aplicação da mineração de dados na IoT



Fonte: adaptada de Gupta e Gupta (2017).

Armazém de dados (do inglês Data Warehouse) é uma estratégia de organização de dados voltada para a gestão estratégica da empresa (CALDEIRA, 2012). Nesse sentido, um armazém de dados possibilita gerir um grande volume de dados que provém de diversas fontes espalhadas dentro de uma organização. Assim, através da consolidação e análise desses dados de diferentes fontes, é possível extrair informações que permitem posicionar

estrategicamente a empresa. Para garantir que os armazéns de dados possam consolidar os dados gerados pelas diferentes aplicações da IoT, é necessário que eles sejam adaptados para lidar com dados não estruturados, heterogêneos, em tempo real e, algumas vezes, incertos, que são gerados pelas aplicações IoT.

Armazém de dados e mineração de dados, apesar de, normalmente, aparecerem em um mesmo texto, representam duas distintas tecnologias. Realizar a correta distinção entre esses conceitos é de extrema importância para realizar uma análise eficiente dos dados. Isso ocorre, pois o armazém de dados e a mineração de dados integram etapas distintas no processo de análise de dados. O Quadro 3.5 mostra uma comparação entre cada um desses termos.

Quadro 3.5 | Comparação entre os termos armazém de dados e mineração de dados

Armazém de Dados	Mineração de Dados
É um processo utilizado para integrar dados de diferentes fontes em um mesmo banco de dados.	É um processo utilizado para extrair padrões e relacionamento entre os dados.
Proporciona, para as organizações, um mecanismo para armazenamento de um grande volume de dados.	É uma técnica que é aplicada sobre os armazéns de dados.
Esse processo deve ser realizado em uma fase anterior à mineração de dados.	É um processo que é realizado após os dados terem sido consolidados através do armazém de dados.
Repositório para armazenamento de dados.	Atividade realizada para encontrar padrões e dar significado aos dados.

Fonte: adaptado de EDUCBA (2019).

Pesquise mais

Consulte um excelente livro que apresenta alguns conceitos e exemplos de aplicações da análise de dados sobre o Big Data da IoT e alguns estudos de caso sobre a união dessas tecnologias. Nesta obra existe, por exemplo, um projeto de um sistema de automação residencial que utiliza um robô móvel para interação entre o sistema e os usuários. Nesse projeto estão disponíveis os códigos e a estrutura funcional de todo o sistema.

SOLANKI, Vijender Kumar; KUMAR, Raghvendra; KHARI, Manju. **Internet of Things and Big Data Analytics for Smart Generation**. Springer, 2019.

Como pode ser visto pelo Quadro 3.2 apesar de, normalmente, estarem em um mesmo contexto os conceitos de mineração de dados e armazém de dados são bastante diferentes. É através da realização da mineração de dados sobre o armazém de dados que é possível encontrar padrões e definir

tendências para as organizações. De posse dessas informações extraídas dos conjuntos de dados é que são gerados os relatórios, gráficos e indicadores que possibilitam traçar os caminhos para investimento e gerar valor ao negócio.

Você aprendeu como os dados gerados pela IoT podem acarretar a melhoria dos produtos, serviços e processos. Além disso, você aprendeu as etapas necessárias para extrair valor dos dados gerados pela IoT. Com todo esse conhecimento adquirido agora você é capaz de construir aplicações que sejam empregadas na resolução de problemas reais e que promovam a melhoria do bem-estar de toda a sociedade. Agora você está preparado para mudar o mundo. Então, mãos à obra!

Sem medo de errar

A análise dos dados gerados pelas aplicações da IoT correspondem a um dos principais objetivos a serem alcançados pela IoT. Isso ocorre, pois é através da análise efetiva desses dados que é possível gerar valor de negócio a partir das aplicações IoT. Assim, construir uma aplicação que possibilite a integração com as plataformas em nuvem e realização da análise de dados é requisito fundamental para as aplicações da IoT.

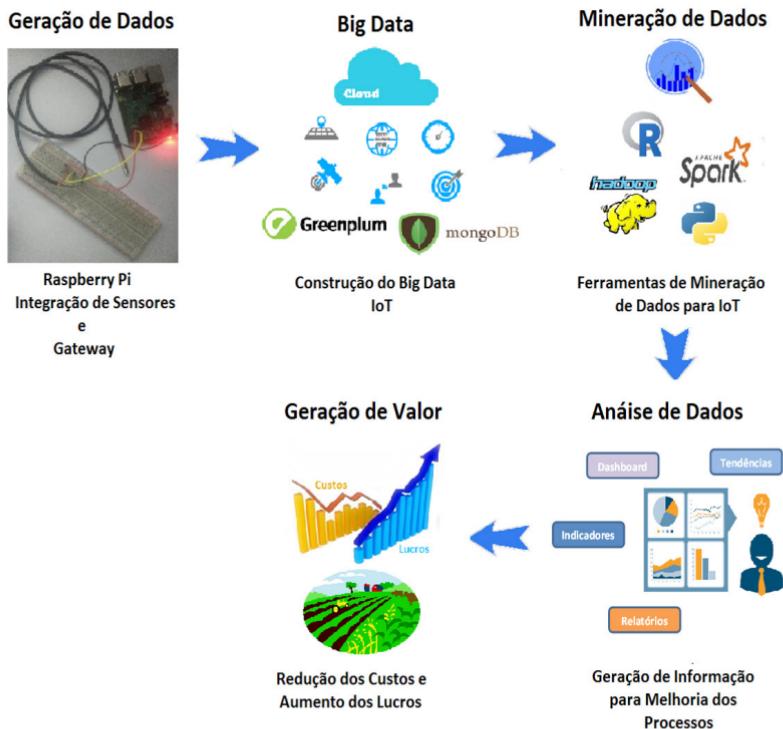
Você está trabalhando na construção de uma aplicação que permita realizar o monitoramento climático de uma grande lavoura de soja. Essa aplicação corresponde a um instrumento inteligente conectado à internet e que possibilita a visualização remota das variáveis de interesse e o emprego de algoritmos de mineração de dados. Esses algoritmos devem ser aplicados a fim de contribuir com o melhor entendimento do processo produtivo e incremento da qualidade dos produtos.

Nesta última fase do projeto, você deve apresentar um esquema para a construção desse instrumento inteligente. Esse esquema deve conter as etapas necessárias para a realização eficiente dessa análise. O protótipo desse instrumento é construído utilizando o Rpi e os dados coletados devem ser enviados à internet a fim de permitir a visualização remota das variáveis medidas pelos sensores e a aplicação dos algoritmos de mineração de dados.

A Figura 3.22, apresenta um modelo de protótipo para o desenvolvimento do instrumento inteligente utilizando o Raspberry Pi e o esquema para geração de valor através desse instrumento. Além disso, nessa figura, também está presente as ferramentas de software a serem empregadas, como esse protótipo deve ser conectado a uma plataforma de computação em nuvem e quais as etapas necessárias para a geração de valor de negócio através desse instrumento. Pela Figura 3.13, é possível verificar que o Rpi é utilizado como plataforma de integração de sensores e gateway para

disponibilização dos dados na internet. Desse modo, o Rpi pode ser visto como uma ferramenta importante no desenvolvimento de aplicações IoT. Após a integração dos diferentes dados, eles podem ser enviados para as plataformas em nuvem. Como esses dados são, geralmente, não estruturados é necessário utilizar bancos de dados não relacionais como o mongoDB. Aos dados gerados pelo instrumento inteligente, podem ser adicionados dados gerenciais como expectativa de investimento e retorno esperado, o que torna o Big Data ainda mais valioso para a tomada de decisão. Após serem tratados, podem ser aplicadas ferramentas de mineração de dados. As ferramentas mais comuns que permitem visualização e aplicação de algoritmos de aprendizado de máquina são o Python e o R. O Apache Spark e o Hadoop também auxiliam nessa tarefa, pois podem tratar grandes massas de dados de maneira distribuída. Após essa etapa pode ser gerado o conhecimento sobre o processo produtivo. É através desse conhecimento que podem ser tomadas as decisões que promovem a redução dos custos de produção, melhoria da qualidade dos produtos e maior geração de lucros.

Figura 3.22 | Esquema para construção do instrumento inteligente



Fonte: elaborada pelo autor.

Os dados para essa aplicação devem ser analisados, inicialmente, em tempo real, pois as mudanças de umidade, qualidade do solo e contaminação por pragas necessitam de medidas corretivas imediatas. Caso essas medidas não sejam tomadas em tempo hábil, toda a produção pode ser comprometida.

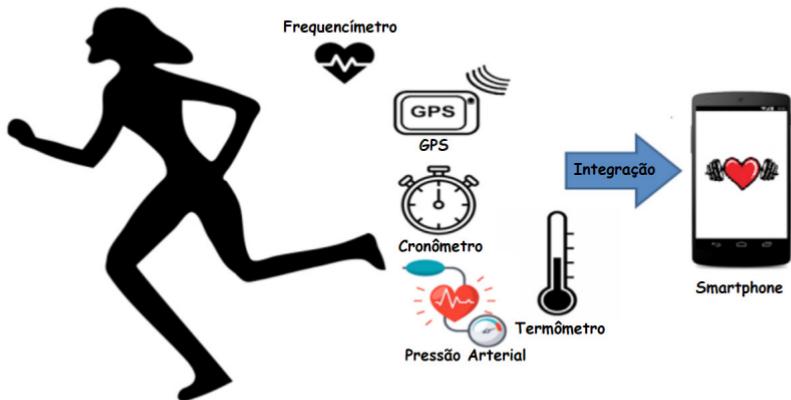
Para realizar essa tarefa de análise em tempo real podem ser empregados algoritmos de aprendizado de máquina como as redes neurais artificiais. Essas redes, após o processo de treinamento, podem identificar qual a combinação dos dados (umidade do solo, temperatura e qualidade do solo) correspondem a um comportamento normal. Assim, quando novos dados coletados pelos sensores são recebidos, é possível reconhecer um comportamento que esteja fora do padrão habitual. Desse modo, podem ser tomadas medidas para resolver esses problemas antes que danos maiores sejam causados à produção.

Avançando na prática

Monitoramento das atividades físicas de atletas

Devido ao sucesso alcançado pelo instrumento inteligente, a empresa onde você trabalha passou a receber solicitações de desenvolvimento de outros instrumentos inteligentes para o monitoramento de várias aplicações. Dentro dessas aplicações, você passou a ser o responsável pelo desenvolvimento de um sensor inteligente que permite o monitoramento das atividades e das condições físicas dos atletas. Para isso, já foram definidos os sensores a serem utilizados. Devem ser empregados medidores de frequência cardíaca, pressão arterial, termômetro, GPS, para medir a distância percorrida, e um monitor de tempo da atividade executada. Esses dados devem ser consolidados através de smartphones e têm de ser enviados para plataformas em nuvem. A análise desses dados deve permitir que sejam criados treinos específicos para cada atleta a fim de melhorar o desempenho na atividade executada. O relatório inicial já foi desenvolvido, agora você precisa fornecer um esquema que mostre como esses dados devem ser consolidados, analisados e quais ferramentas devem ser utilizadas. A Figura 3.23 apresenta um exemplo dessa aplicação.

Figura 3.23 | Aplicação de monitoramento de atletas

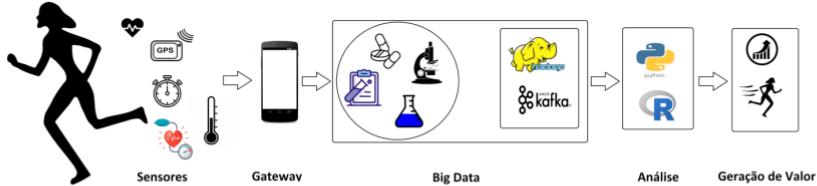


Fonte: elaborada pelo autor.

Resolução da situação-problema

A Figura 3.24 apresenta um possível esquema para consolidação e análise dos dados de diferentes atletas através das plataformas em nuvem. Por meio dessa figura é possível verificar que para a consolidação dos dados foi utilizado o smartphone e que esses dados podem ser analisados de maneira off-line, pois serão utilizados para monitorar o desempenho do atleta e sugerir treinos específicos que permitam uma melhor performance desse atleta.

Figura 3.24 | Exemplo de esquema para consolidação e análise dos dados de atletas



Fonte: elaborada pelo autor.

Os dados gerados pelos diferentes sensores podem ser enviados para o smartphone, por exemplo, através do protocolo Bluetooth. No smartphone, pode existir um aplicativo que executa um processamento inicial desses dados. Esse aplicativo pode, por exemplo, através da aplicação de algoritmos de mineração de dados, identificar se existe algum sensor que não esteja funcionando corretamente. Do smartphone, esses dados podem ser enviados para plataformas em nuvem. Nas plataformas em nuvem, os dados sobre a

atividade física de um atleta podem ser comparados com outros atletas de uma mesma modalidade. Além disso, aos dados dessa atividade, podem ser adicionados outros dados como os calendários de competições, períodos de descanso e exames laboratoriais mais complexos. Essa união de dados de várias diferentes fontes visa tornar as decisões mais precisas. De posse desses dados, podem ser aplicadas as técnicas de análise de séries temporais para realizar previsões de evolução nos treinos e algoritmos de aprendizado de máquina para identificar, por exemplo, possibilidades de lesões. Assim, é possível analisar os dados de vários atletas, encontrar as deficiências e ajustar os treinos visando um melhor desempenho e adequação das atividades ao calendário de competições.

Faça valer a pena

1. Mineração de dados e armazém de dados são conceitos muito importantes para a compreensão e aplicação da análise de dados sobre o Big Data, especialmente, quando esses dados provêm da IoT. A distinção entre esses conceitos é fundamental nessa análise. De acordo com as informações apresentadas no quadro a seguir, faça a associação entre os conceitos e suas características.

Coluna A	Coluna B
Conceitos	Características
1. Mineração de dados.	I. Utilizado para extrair padrões e relacionamento entre dados.
	II. Utilizado para integração de dados.
2. Armazém de dados.	III. Utilizado para o armazenamento de um grande volume de dados.
	IV. Utilizado sobre os dados já armazenados.

Assinale a alternativa que apresenta a associação CORRETA entre as colunas.

- a. I – 1; II – 1; III – 2; IV – 2.
- b. I – 1; II – 2; III – 1; IV – 2.
- c. I – 2; II – 2; III – 2; IV – 1.
- d. I – 1; II – 2; III – 2; IV – 1.
- e. I – 2; II – 2; III – 1; IV – 1.

2. A análise de dados é realizada para que seja possível transformar os dados coletados pelos sensores da IoT em informações que possam ser empregadas a fim de melhorar o entendimento sobre os processos e a qualidade de produtos. A seguir estão listadas algumas afirmações sobre o processo de análise de dados realizados pela IoT.

- I. A escalabilidade para a análise dos dados coletados é um dos principais elementos a serem contemplados para assegurar uma aplicação eficiente da IoT.
- II. A mineração de dados é um dos exemplos utilizados para a análise de dados da IoT.
- III. A integração entre a IoT e as plataformas em computação em nuvem possibilita a aplicação de estratégias como o aprendizado de máquina na análise dos dados.
- IV. Os armazéns de dados são utilizados como algoritmos para encontrar padrões e relacionamentos entre os dados da IoT.

Indique qual das alternativas a seguir apresenta as afirmações corretas sobre a análise de dados da IoT.

- a. Somente I e III estão corretas.
- b. Somente I, II e IV estão corretas.
- c. Somente I e III estão corretas.
- d. Somente I, II e III estão corretas.
- e. Todas as afirmações estão corretas.

3. Para agregar valor ao negócio a partir dos dados gerados pelas aplicações IoT é necessário que algumas etapas sejam seguidas. Essa sequência de etapas se inicia com a coleta de dados realizada pelos sensores da IoT e termina com a tomada de decisão, que é sustentada pela análise desses dados.

Escolha, dentre as alternativas a seguir, qual apresenta as características corretas sobre a análise de dados da IoT.

- a. A análise de dados em aplicações IoT envolve a consolidação dos dados e a aplicação de algoritmos específicos que possam lidar com as peculiaridades do Big Data IoT.
- b. A análise de dados em aplicações IoT é realizada através do processamento local, sem a necessidade de utilizar o processamento remoto.

- c. O Big Data da IoT fornece dados que não podem ser utilizados como entrada de algoritmos de mineração de dados, pois as informações geradas são incorretas.
- d. A escalabilidade para a análise do Big Data da IoT não representa uma questão a ser observada, pois como existem milhões de dispositivos conectados à internet, pode existir uma troca de mensagens entre esses dispositivos, o que garante uma menor complexidade independentemente da quantidade de dispositivos.
- e. As peculiaridades do Big Data da IoT não representam um empecilho à aplicação dos algoritmos tradicionais de mineração de dados, pois devido à capacidade ilimitada de processamento gerada pelas plataformas em nuvem, essas características podem ser superadas.

Referências

- ABBASI, A. Z. *et al.* A review of wireless sensors and networks' applications in agriculture. **Computer Standards & Interfaces**, v. 36, n. 2, p. 263-270, 2014.
- ALAMRI, A. *et al.* A survey on sensor-cloud: architecture, applications, and approaches. **International Journal of Distributed Sensor Networks**, v. 9, n. 2, p. 917923, 2013.
- ALEMDAR, H.; ERSOY, C. Wireless sensor networks for healthcare: A survey. **Computer networks**, v. 54, n. 15, p. 2688-2710, 2010.
- ATLAM, H. F. *et al.* Integration of cloud computing with internet of things: challenges and open issues. In: **2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)**. IEEE, 2017. p. 670-675.
- BEAGLEBOARD.ORG. **Getting Started**. Página Inicial. Disponível em: <https://beagleboard.org/bone>. Acesso em: 26 ago. 2019.
- BIRMINGHAM, W. P.; SIEWIOREK, D. P. MICON: a knowledge based single board computer designer. In: **21st Design Automation Conference Proceedings**. IEEE, 1984. p. 565-571.
- BISWAS, A. R.; GIAFFREDA, R. IoT and cloud convergence: Opportunities and challenges. In: **2014 IEEE World Forum on Internet of Things (WF-IoT)**. IEEE, 2014. p. 375-376.
- BOTTA, A. *et al.* Integration of cloud computing and internet of things: a survey. **Future generation computer systems**, v. 56, p. 684-700, 2016.
- BRILL, E.; MOORE, R. C. An improved error model for noisy channel spelling correction. In: **Proceedings of the 38th Annual Meeting on Association for Computational Linguistics**. Association for Computational Linguistics, 2000. p. 286-293.
- BYUN, J. *et al.* An intelligent self-adjusting sensor for smart home services based on ZigBee communications. **IEEE Transactions on Consumer Electronics**, v. 58, n. 3, p. 794-802, 2012.
- CALDEIRA, C. P. **Data Warehousing**: Conceitos e Modelos. Edições Sílabo, 2012.
- CIRCUIT BASICS. **RASPBERRY PI DS18B20 TEMPERATURE SENSOR TUTORIAL**. [S.I.], 2016. Disponível em: <http://www.circuitbasics.com/raspberry-pi-ds18b20-temperature-sensor-tutorial/>. Acesso em: 26 ago. 2019.
- CROFT, W. B.; METZLER, D.; STROHMAN, T. **Search engines**: Information retrieval in practice. Reading: Addison-Wesley, 2010.
- DARWISH, A.; HASSANIEN, A. E. Wearable and implantable wireless sensor network solutions for healthcare monitoring. **Sensors**, v. 11, n. 6, p. 5561-5595, 2011.
- DINH, T.; KIM, Y. An efficient interactive model for on-demand sensing-as-a-services of sensor-cloud. **Sensors**, v. 16, n. 7, p. 992, 2016.

EDUCBA. Data Warehousing VS Data Mining – 4 Awesome Comparisons. Disponível em: <https://www.educba.com/data-warehousing-vs-data-mining/>. Acesso em: 29 ago. 2019.

ELEMENT14. Raspberry Pi 3 Model B GPIO 40 Pin Block Pinout. [S.I.], 2016. Disponível em: <https://www.element14.com/community/docs/DOC-73950/l/raspberry-pi-3-model-b-gpio-40-pin-block-pinout>. Acesso em: 05 jun. 2019.

EMMERSON, B. M2M: the Internet of 50 billion devices. **WinWin Magazine**, v. 1, p. 19-22, 2010.

ERL, T.; PUTTINI, R.; MAHMOOD, Z. **Cloud computing: concepts, technology & architecture**. Pearson Education, 2013.

FEDRIZZI, M.; SORIA, J. Application of a single-board computer as a low-cost pulse generator. **Measurement Science and Technology**, v. 26, n. 9, p. 095302, 2015.

FERNANDO, T. **How to build a Raspberry Pi thermometer you can access anywhere (a beginner's guide)**. [S.I.], 2016. Disponível em: <https://blog.dataplicity.com/how-to-build-a-raspberry-pi-thermometer-you-can-access-anywhere-a-beginner-s-guide-4ad44ce9f4c9>. Acesso em: 29 ago. 2019.

FORTINO, G.; TRUNFIO, P. (ed.). **Internet of things based on smart objects**: Technology, middleware and applications. Springer Science & Business Media, 2014.

GHAYVAT, H. *et al.* WSN-and IOT-based smart homes and their extension to smart buildings. **Sensors**, v. 15, n. 5, p. 10350-10379, 2015.

GRINBERG, M. **Flask web development**: developing web applications with python. “ O'Reilly Media, Inc”, 2018.

GROUP, Sap Data Science. Data Science and Machine Learning in the Internet of Things and Predictive Maintenance. 2017. Disponível em: <https://www.sap.com/documents/2016/10/8ec7f-23f-917c-0010-82c7-eda71af511fa.html>. Acesso em: 29 ago. 2019.

GUNICORN.ORG. **Installation**. Página inicial, 2017. Disponível em: <https://gunicorn.org/>. Acesso em: 29 ago. 2019.

GUPTA, P.; GUPTA, R. Data mining framework for IoT applications. **International Journal of Computer Applications**, v. 174, n. 2, p. 4-7, 2017.

HANES, D. *et al.* **IoT fundamentals: Networking technologies, protocols, and use cases for the internet of things**. Cisco Press, 2017.

HANG, L. *et al.* Design and Implementation of a Sensor-Cloud Platform for Physical Sensor Management on CoT Environments. **Electronics**, v. 7, n. 8, p. 140, 2018.

HARRINGTON, W. **Learning Raspbian**. Packt Publishing Ltd, 2015.

HOCKER III, L. O. **UART protocol that provides predictable delay for communication between computers of disparate ability**. U.S. Patent n. 5,682,508, 28, out. 1997.

INTEL. **Introdução às Placas Intel Galileo.** [S.l., s.d.]. Disponível em: <https://www.intel.com.br/content/www/br/pt/support/articles/000005912/boards-and-kits/intel-galileo-boards.html>. Acesso em: 26 ago. 2019.

ISIKDAG, U. Enhanced building information models. **Enhanc. Build. Inf. Model. Using IoT Serv. Integr.** Patterns, p. 13-24, 2015(b).

ISIKDAG, U. Internet of Things: Single-board computers. In: **Enhanced Building Information Models**. Springer, Cham, 2015(a). p. 43-53.

JAIN, A. D. S. et al. Application of big data in supply chain management. **Materials Today: Proceedings**, v. 4, n. 2, p. 1106-1115, 2017.

JAIN, A. K.; LI, S. Z. **Handbook of face recognition**. New York: springer, 2011.

JINDAL, N.; LIU, B. Review spam detection. In: **Proceedings of the 16th international conference on World Wide Web**. ACM, 2007. p. 1189-1190.

KITCHIN, R.; MCARDLE, G. What makes Big Data, Big Data? Exploring the ontological characteristics of 26 datasets. **Big Data & Society**, v. 3, n. 1, p. 2053951716631130, 2016.

KURNIAWAN, A. **Smart Internet of Things Projects**. Packt Publishing Ltd, 2016.

LEMOS, M. et al. Uma abordagem para provisionamento automático de sensores em nuvens de sensores. **Learning and Nonlinear Models**, 14. 4-14. 10.21528/LNLM-vol14-no1-art1, 2016.

LI, H.; OTA, K.; DONG, M. Learning IoT in edge: deep learning for the internet of things with edge computing. **IEEE Network**, v. 32, n. 1, p. 96-101, 2018.

LÓPEZ, G.; QUESADA, L.; GUERRERO, L. A. Alexa vs. Siri vs. Cortana vs. Google assistant: a comparison of speech-based natural user interfaces. In: **International Conference on Applied Human Factors and Ergonomics**. Springer, Cham, 2017. p. 241-250.

MAJUMDER, S. et al. Smart homes for elderly healthcare—Recent advances and research challenges. **Sensors**, v. 17, n. 11, p. 2496, 2017.

MAKSIMOVIĆ, M. et al. Raspberry Pi as Internet of things hardware: performances and constraints. **Design issues**, v. 3, n. 8, 2014.

MANEA, G.; POPA, S. Integration of sensor networks in cloud computing. **UPB Sci. Bull., series C**, v. 78, n. 2, 2016.

MARJANI, M. et al. Big IoT data analytics: architecture, opportunities, and open research challenges. **IEEE Access**, v. 5, p. 5247-5261, 2017.

MELL, P.; GRANCE, T. The NIST definition of cloud computing: Recommendations of the National Institute of Standards and Technology. In: **NIST Special Publication 800-145**. Computer Security Division, 2011. Disponível em: <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>. Acesso em: 26 ago. 2019.

MENZE, M.; GEIGER, A. Object scene flow for autonomous vehicles. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**, 2015. p. 3061-3070.

MINTEER, A. **Analytics for the Internet of Things (IoT): Intelligent analytics for your intelligent devices**. Birmingham: Packt Publishing, . 369 p.

MJROBOT. **Python WebServer With Flask and Raspberry Pi**. Publicado em 5 jan. 2018. Disponível em: <https://www.hackster.io/mjrobot/python-webserver-with-flask-and-raspberry-pi-41b5fc>. Acesso em: 26 ago. 2019.

PERUMAL, B. et al. WSN integrated cloud for automated telemedicine (ATM) based e-health-care applications. In: **Proceedings of the 4th International Conference on Bioinformatics and Biomedical Technology (IPCBEE'12)**. 2012. p. 166-170.

RASPBERRY PI 3 - instalando sistema operacional. [S.L.: s.n.], 2016. 1 vídeo (38:53 min). Publicado pelo canal Projeto Root. Disponível em: <https://www.youtube.com/watch?v=u0PP-fg0PXNc>. Acessos em: 27 ago. 2019.

RASPBERRY PI. **About Us**. Página inicial, [s.d.]. Disponível em: <https://www.raspberrypi.org/>. Acesso em: 26 ago. 2019.

RASPBERRY PI. **Getting started with picamera**. [S.I.], 2016. Disponível em: <https://projects.raspberrypi.org/en/projects/getting-started-with-picamera>. Acesso em: 26 ago. 2019.

REDMON, J. et al. You only look once: Unified, real-time object detection. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**, 2016. p. 779-788.

SAMUEL, A. Some Studies in Machine Learning Using the Game of Checkers. **IBM Journal of Research and Development**, v. 3, n. 3, p. 210-29, Jul. 1959.

SCHMIDT, E. **Conversation with Eric Schmidt hosted by Danny Sullivan**. Google Press Center, 2006. Disponível em: <https://www.google.com/press/podium/ses2006.html>. Acesso em: 26 ago. 2019.

SEMICONDUCTOR, D. **DS18B20 Datasheet**. [S.I.], 2017. Disponível em: <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>. Acesso em: 26 ago. 2019.

SENTHILKUMAR, G.; GOPALAKRISHNAN, K.; KUMAR, V. S. Embedded image capturing system using raspberry pi system. **International Journal of Emerging Trends & Technology in Computer Science**, v. 3, n. 2, p. 213-215, 2014.

SOLDATOS, J. (Ed.). **Building Blocks for IoT Analytics**. River Publishers, 2016.

SU, X. et al. **Introduction to Big Data**. Learning material is developed for course IINI3012 Big Data, v. 2, 2018.

SURYADEVARA, N. K. et al. WSN-based smart sensors and actuator for power management in intelligent buildings. **IEEE/ASME transactions on mechatronics**, v. 20, n. 2, p. 564-571, 2014.

TAN, K. What's next?: Sensor+ cloud!?. In: **Proceedings of the Seventh International Workshop on Data Management for Sensor Networks**. ACM, 2010.

THE PALLETS PROJECTS. **Flask**. [S.l., s.d.]. Disponível em: <https://palletsprojects.com/p/flask/>. Acesso em: 26 ago. 2019.

TONGKE, F. Smart agriculture based on cloud computing and IOT. **Journal of Convergence Information Technology**, v. 8, n. 2, 2013.

TRUONG, H. et al. On engineering analytics for elastic IoT cloud platforms. In: **International Conference on Service-Oriented Computing**. Springer, Cham, 2016. p. 267-281.

UDOOL. **Start here**. Página inicial, [s.d.]. Disponível em: <https://www.udoo.org/getting-started/>. Acesso em: 26 ago. 2019.

WEN, C. L. Telemedicina e telessaúde: um panorama no Brasil. **Informática Pública**, v. 10, n. 2, p. 7-15, 2008.

WHAT IS Arduino?. **Arduino**. [S.l., s.d.]. Disponível em: <https://www.arduino.cc/en/Guide/Introduction>. Acessos em: 26 ago. 2019.

ZHONG, Y. et al. A vision-based counting and recognition system for flying insects in intelligent agriculture. **Sensors**, v. 18, n. 5, p. 1489, 2018.

ZHOU, J. et al. Cloudthings: A common architecture for integrating the internet of things with cloud computing. In: **Proceedings of the 2013 IEEE 17th international conference on computer supported cooperative work in design (CSCWD)**. IEEE, 2013. p. 651-657.

Unidade 4

Novos desenvolvimentos em IoT

Convite ao estudo

Você já parou para pensar que a Internet das Coisas (IoT) já faz parte do nosso dia a dia? Pulseiras inteligentes utilizadas para monitorar a qualidade do sono, relógios inteligentes que podem ser empregados para o acompanhamento de exercícios físicos, a previsão de consumo de energia, o rastreamento inteligente de entregas e a integração de diferentes dispositivos para o monitoramento de residências e edifícios são exemplos da IoT em nossas vidas. Tudo isso provém da ideia de integrar diferentes tecnologias de sensoriamento, atuação e comunicação para criar objetos que sejam capazes de interagir com o ambiente. Essas transformações geradas por essa ideia proporcionam o desenvolvimento de melhores produtos e maior comodidade aos usuários. Quando essas transformações são levadas para os setores industriais e de serviços, o impacto transformador da IoT é ainda maior. Isso ocorre, pois, por meio da IoT, esses setores podem reduzir custos, melhorar a qualidade e a eficiência dos produtos e processos, além de proporcionar o desenvolvimento de aplicações que possam mudar a vida das pessoas.

Com toda a revolução provocada pela IoT, é necessário conhecer algumas de suas aplicações práticas e os impactos gerados por essas transformações. Com esse conhecimento é possível compreender a dimensão da revolução provocada por esse novo paradigma. Como resultado desse aprendizado, podemos desenvolver soluções tecnológicas que auxiliem na resolução de problemas reais.

Para ajudar na tarefa de construir soluções para problemas reais, nesta unidade, você conhecerá os conceitos de computação em neblina e de borda. Essas estratégias foram criadas para contornar alguns problemas inerentes ao processo de conexão dos dispositivos IoT com as plataformas em nuvem. Problemas como latência e largura de banda podem ser contornados com a aplicação desses conceitos. Além deles, você conhecerá as tecnologias empregadas para a construção de veículos conectados e como eles realizam a comunicação e a integração com diferentes sistemas de transporte inteligente. Outras aplicações práticas da IoT que serão apresentadas nesta seção são a automação industrial, o agronegócio e os serviços de saúde. O conhecimento dessas aplicações é fundamental para compreender o funcionamento de um sistema que emprega os conceitos de “em todo tempo”, “em todo lugar” e “em todas as coisas” presentes na IoT. Ao final desta seção, você também

aprenderá como desenvolver um plano de negócios para criar ou gerenciar a construção de uma aplicação IoT.

Agora, imagine que você foi contratado por uma grande rede hospitalar para desenvolver um sistema capaz de realizar o acompanhamento remoto de pacientes. O acompanhamento deve ser integrado com uma central de monitoramento onde existem médicos especialistas e sistemas de emergência. Quando um comportamento anormal nos dados coletados pelos sensores “vestíveis” (do inglês *wearable*) instalados no paciente e na residência é identificado, automaticamente, deve ser tomada alguma medida que garanta a melhor assistência a pacientes monitorados. As medidas podem ser, por exemplo, acionar os serviços de emergência para recolhimento do paciente ou indicar uma visita a um centro de saúde mais próximo. O sistema também deve estar integrado aos serviços de emergência e de trânsito para reconhecimento de melhores rotas a fim de facilitar o resgate dos pacientes com maior urgência de atendimento. Ao final desse projeto, os diretores dessa grande rede hospitalar desejam receber um relatório contendo o plano de negócios para essa aplicação, todos os sistemas de hardware e software a serem utilizados e os esquemas de conexão e interligação de todos os componentes.

Agora que você já tem conhecimento para desenvolver aplicações que possam mudar as vidas das pessoas, chegou a hora de construir um mundo melhor, em que o acesso a serviços de qualidade pode estar ao alcance de todas as pessoas. Seja parte dessa transformação e utilize o conhecimento adquirido para fazer a diferença e tornar a nossa sociedade mais justa e eficiente.

Bons estudos!

Seção 1

Computação em neblina e computação de borda (nuvens locais)

Diálogo aberto

As possibilidades geradas pela conexão dos objetos físicos do nosso cotidiano por meio da internet são enormes. Com eles, é possível monitorar praticamente qualquer atividade realizada. A partir desse monitoramento, podem ser obtidos dados que possibilitam a análise e o melhor entendimento do ambiente. Por exemplo, no acompanhamento e na previsão de possível colapso em grandes obras de engenharia. Em grandes estruturas, como pontes, barragens de rejeitos de minérios, hidrelétricas e viadutos, podem ser adicionados diferentes tipos de sensores inteligentes que estejam conectados à internet e enviando os dados para serem analisados em tempo real. A análise dos dados permite a identificação de comportamentos anormais que possam ser utilizados para prever o colapso da estrutura. Assim, problemas podem ser evitados e muitas vidas, preservadas.

Quando esses dados são coletados e prontamente enviados para as plataformas em nuvem sem nenhum tratamento prévio, alguns problemas, como o uso excessivo da banda de transmissão e atrasos no recebimento de respostas, tendem a aparecer. Especialmente em aplicações que demandam o acompanhamento em tempo real, esses problemas podem ter consequências desastrosas. Por exemplo, em grandes obras de engenharia, como barragens de rejeito de minério, as anomalias estruturais precisam ser rapidamente identificadas e medidas corretivas devem ser tomadas no menor intervalo de tempo possível. Essa ação rápida é um fator primordial para o sucesso da aplicação. Isso ocorre, pois o tempo entre a identificação de um possível colapso e a realização efetiva de ações que possam evitar uma catástrofe é normalmente muito curto. Para contornar esses problemas de atraso entre a requisição de um serviço e a execução da tarefa, foram desenvolvidos os conceitos de computação em neblina e de borda. Com a computação em neblina também é possível reduzir os problemas causados pela excessiva quantidade de mensagens trocadas entre os sensores e a nuvem.

Nesta seção, você foi contratado para desenvolver uma aplicação que integre diferentes sistemas (hospitalar, de transporte e de emergência) para realizar o monitoramento remoto de pacientes. Nesta primeira fase, você deve entregar um relatório que contenha a indicação dos sensores, dos atuadores e dos Single Board Computer utilizados. Além disso, esse relatório deve conter

o esquema de conexão entre cada um dos componentes desse sistema. Para isso, você deve empregar os conceitos de computação em nuvem que possibilitem o monitoramento em tempo real do paciente.

Através dos exemplos de aplicações e dos conceitos apresentados nesta seção será possível construir o relatório para a conclusão dessa primeira etapa de elaboração do projeto de monitoramento de pacientes. Ao final desta seção, você será capaz de adicionar a computação em neblina aos seus projetos e de torná-los aptos a serem empregados para a solução de problemas em tempo real. Então, vamos começar?

Não pode faltar

Desde o surgimento do termo Internet das Coisas, muitas pesquisas e esforços têm sido empregados para tornar essa tecnologia cada vez mais confiável e abrangente. A utilização, por exemplo, da computação em nuvem possibilitou a extensão virtual da capacidade de armazenamento e processamento dos dispositivos (nós) da IoT. Por meio da virtualização desses dispositivos é possível que os dados gerados sejam armazenados, visualizados e analisados por meio das plataformas de computação em nuvem.

Após a análise dos dados, pode ser realizada a interação com o ambiente, por exemplo, na aplicação de monitoramento de terremotos (MICROSOFT, 2017). Sensores são responsáveis por monitorar as alterações de algumas variáveis relacionadas aos terremotos e enviar os dados para plataformas em nuvem. Após a agregação desses dados é possível identificar os padrões de comportamentos das condições do ambiente. Caso seja identificado algum comportamento que indique a iminência de um terremoto, as plataformas em nuvem podem enviar, automaticamente, mensagens de alerta para as redes de sensores e os atuadores podem, por exemplo, disparar alarmes para alertar a população. A velocidade na identificação do terremoto e na tomada de decisão é fundamental para o sucesso dessa aplicação. Isso ocorre, pois o tempo necessário para tomar uma decisão deve ser inferior ao intervalo de tempo entre a ocorrência de um terremoto e a propagação das ondas sísmicas até as áreas habitadas. Esse intervalo de tempo entre um sensor coletar um dado e ele ser recebido pela plataforma em nuvem é conhecido como latência (SMOOT, TAN, 2011).

Assim, quando a latência é um fator fundamental para o êxito de uma aplicação, garantir que esse requisito seja atendido passa a ser um dos grandes obstáculos para as arquiteturas IoT (SCHULZ *et al.*, 2017). Em aplicações de tempo real, como a supervisão de pacientes, o controle de

processos industriais e o monitoramento de veículos, latências excessivas podem ocasionar um colapso do sistema.

Um exemplo de sistema que necessita de dados em tempo real é o ABS (Anti-lock Braking System). Esse sistema precisa receber, em tempo real, as informações sobre a circunstância de condução a que o automóvel está sujeito e, em frações de segundos, tomar a decisão assertiva para evitar o travamento das rodas do veículo (WU; SHIH, 2003). Em situações em que a latência é primordial, o modelo de comunicação direta com plataformas em nuvem não apresenta resultados satisfatórios. Isso ocorre, pois a distância entre o local onde os dispositivos da IoT estão inseridos e os servidores de processamento remoto (plataformas em nuvem) é, normalmente, elevada (TSAI *et al.*, 2014).

Com o aumento do número de dispositivos conectados, a quantidade de dados gerada e armazenada também será maior. Assim, questões sobre o melhor aproveitamento da largura de banda utilizada no envio de dados para a nuvem também devem ser resolvidas para assegurar o desenvolvimento robusto e abrangente da IoT (BONOMI *et al.*, 2012). Para o avanço eficiente da IoT é necessário que sejam adotadas medidas para preencher as lacunas geradas pela latência de comunicação e pelo uso ineficiente da largura de banda.

A fim de preencher essas lacunas, o modelo centralizado de computação em nuvem passa a ser utilizado em conjunto com o paradigma da computação em neblina (do inglês *fog computing*). Na computação em neblina, os dados gerados pelos sensores não são enviados diretamente para as plataformas em nuvem. Por meio da computação em neblina é possível que os dados coletados pelos sensores possam sofrer um primeiro processamento em camadas mais próximas dos dispositivos e dos usuários (BONOMI *et al.*, 2012). Esse primeiro processamento tem como função agrupar dados e, assim, reduzir a quantidade e o tamanho das mensagens enviadas para as plataformas em nuvem. Assim, a realização de um processamento mais próximo dos dispositivos minimiza a largura de banda necessária para a comunicação e reduz o período de latência das requisições.

Um exemplo de aplicação que utiliza a computação em neblina para reduzir a latência e assegurar melhores resultados no monitoramento de pacientes é proposto por Cao *et al.* (2015). Nessa aplicação, em vez de todos os dados sobre a saúde do paciente serem enviados para análise em uma plataforma em nuvem, é realizado um processamento inicial mais próximo do usuário. Esse processamento local dos dados é realizado pelo smartphone do paciente. Adotando a estratégia de computação em neblina foi possível, por exemplo, reduzir os efeitos causados pela queda de pacientes

em decorrência de acidente vascular cerebral. Na computação em neblina, o pré-processamento dos dados ocorre em dispositivos dedicados ou nos gateways das redes locais.

O processamento inicial dos dados também pode ocorrer nos dispositivos que são responsáveis por realizar o monitoramento do ambiente. Quando o pré-processamento desses dados ocorre nos próprios dispositivos, sem a necessidade de serem enviados para outros locais da rede, esse paradigma recebe o nome de computação de borda (SHI *et al.* 2016). Os paradigmas de computação de borda e de neblina são adotados na IoT para o mesmo fim. Ambos são utilizados para reduzir a latência na comunicação e tornar o aproveitamento da largura de banda mais eficiente. A Figura 4.1 apresenta uma representação para aplicações IoT que utilizam os conceitos de computação de borda, computação em neblina e computação em nuvem.

Analisando a Figura 4.1, temos que, conforme ocorre a comunicação entre os dispositivos e a nuvem, a quantidade de elementos diminui, ou seja, existem mais dispositivos responsáveis por realizar a computação de borda do que a computação em neblina. Outra diferença está no fato de que a computação de borda ocorre em locais mais próximos da aplicação. Assim, pode ser visto que a principal diferença entre os conceitos de computação de borda e neblina está no local onde ocorre o pré-processamento dos dados.

Figura 4.1 | Representação da computação em neblina e de borda na IoT



Fonte: adaptada de Tesch (2018).

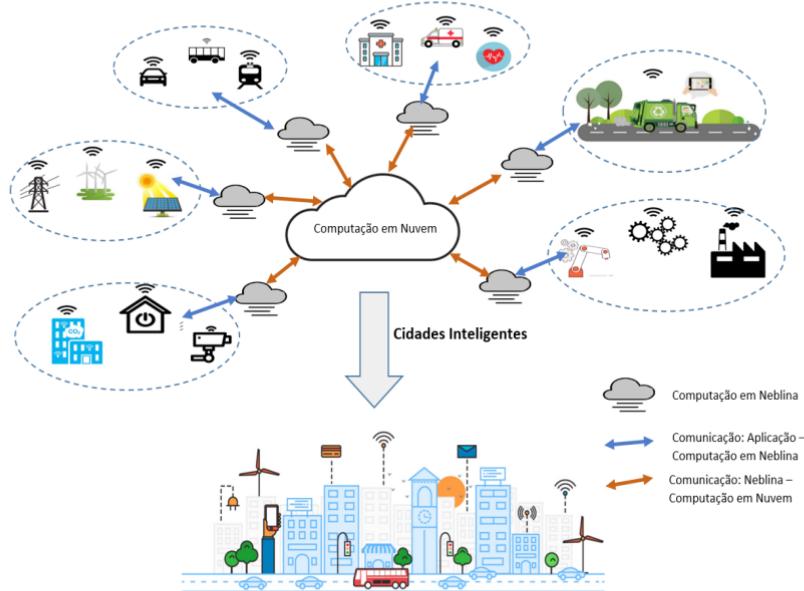
Por meio da união entre os conceitos de computação em neblina e da computação em nuvem, várias aplicações que necessitam que ações sejam realizadas em tempo real podem ser beneficiadas. Além disso, por conta da computação em neblina, as redes da IoT passam a ser mais robustas (PERERA *et al.*, 2017). Isso ocorre, pois a dependência das plataformas em nuvem passa a ser menor, uma vez que algumas das funções realizadas pela nuvem são executadas localmente de maneira distribuída. Dessa forma, mesmo que as conexões com a nuvem sejam interrompidas, as aplicações IoT podem continuar realizando parte das operações de processamento e armazenamento. Várias aplicações podem ser beneficiadas com a união entre a computação em nuvem e a computação em neblina. Entre as aplicações com maior potencial de aproveitamento das oportunidades geradas pela IoT está o desenvolvimento de cidades inteligentes. Por exemplo, podem ser instalados nos ônibus municipais vários sensores para medir a temperatura, a umidade, a intensidade luminosa e a localização. Esses sensores têm a finalidade de coletar informações em diferentes pontos da cidade. Por meio da utilização da computação em neblina, é possível realizar um processamento local dos dados de centenas desses sensores utilizando de maneira eficiente a largura de banda (CRUZ *et al.*, 2018).

Segundo estimativas das Nações Unidas, em 2050, 70% da população mundial estará vivendo nas cidades (UN, 2018). Com a maioria das pessoas vivendo nos centros urbanos, vários problemas enfrentados pelas cidades, como abastecimento de água, energia, engarrafamentos, poluição ambiental, saúde e moradia devem ser sanados para evitar o colapso da infraestrutura urbana. Nesse sentido, utilizar a IoT como meio de promoção do bem-estar da população e da melhoria dos serviços oferecidos passa a ser uma alternativa para o desenvolvimento das cidades e para a preservação do ambiente.

Cidades inteligentes são aplicações que utilizam os conceitos da Internet das Coisas para promover uma forma inteligente de gerenciamento dos diferentes sistemas que compõem as cidades. Sistemas como transporte, saúde, energia, moradia e meio ambiente são integrados pela IoT a fim de proporcionarem serviços mais eficientes e com maior qualidade (GAUR *et al.*, 2015). Além disso, cidades inteligentes também podem ser vistas como a junção de diferentes sistemas inteligentes, por exemplo, o transporte inteligente, o cuidado inteligente com a saúde, o gerenciamento inteligente de energia e do lixo, a segurança e a educação. A integração dessas diferentes aplicações por meio da IoT visa fornecer serviços mais dinâmicos aos habitantes, promover melhor qualidade de vida e construir cidades cada vez mais sustentáveis (LACINÁK; RISTVEJ, 2017).

Como a construção de cidades inteligentes envolve a união de sistemas complexos de monitoramento e gerenciamento da vida urbana, vários sensores e atuadores são alocados em diferentes aplicações. Para aproveitar todas essas possibilidades geradas é necessária a adoção de métodos e técnicas que possam transformar esse conjunto de dados em informações para o gerenciamento efetivo dos recursos oferecidos (LACINÁK; RISTVEJ, 2017). A Figura 4.2 apresenta a representação de alguns sistemas constituintes das cidades inteligentes.

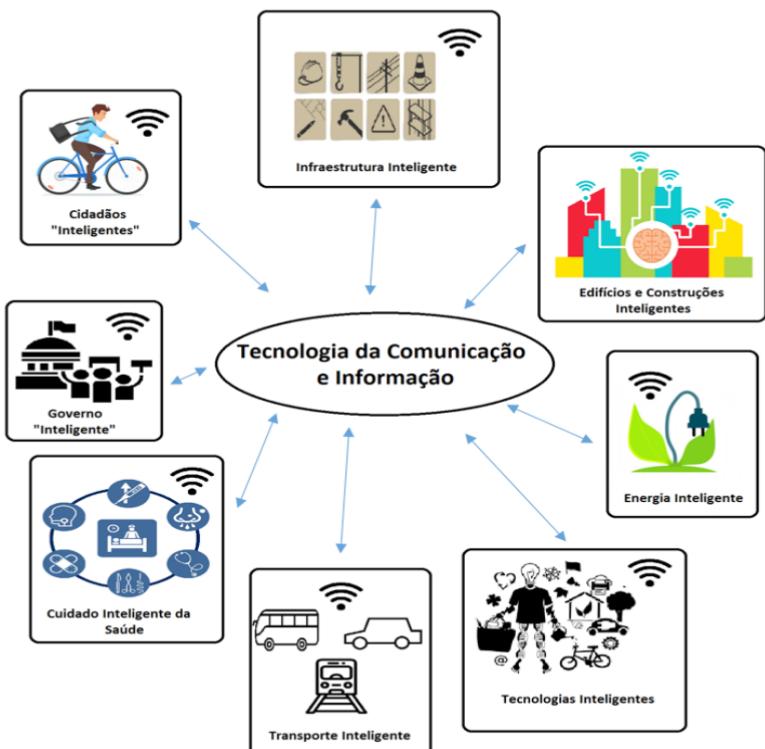
Figura 4.2 | Representação esquemática das cidades inteligentes



Fonte: elaborada pelo autor.

As cidades inteligentes sustentáveis também podem ser vistas como a união de informações e tecnologias de comunicação para proporcionar a melhoria da qualidade de vida dos habitantes, aumentar a eficiência dos serviços urbanos e assegurar as necessidades das gerações presentes e futuras (KONDEPUDI *et al.*, 2014). A Figura 4.3 mostra os oito principais componentes que constituem as cidades inteligentes sustentáveis. Dependendo das características econômicas e das necessidades das cidades, pode ser implementado um conjunto ou apenas alguns desses elementos para a construção das cidades inteligentes sustentáveis (MOHANTY; CHOPPALI; KOUGIANOS, 2016).

Figura 4.3 | Principais componentes das cidades inteligentes



Fonte: adaptada de Mohanty, Choppali e Kougianos (2016).

Um exemplo de sistema que contribui para a construção de cidades cada vez mais sustentáveis e conectadas é a utilização da IoT para o compartilhamento de meios de transporte. O compartilhamento de bicicletas e patinetes elétricos, por exemplo, ajuda a reduzir a poluição e a aliviar o sistema de transporte urbano (LIU, 2018). Os sistemas de compartilhamento de bicicleta e patinetes elétricos utilizam a IoT para permitir que os usuários encontrem a bicicleta ou o patinete mais próximo, que contabilizem o tempo de utilização e que realizem o pagamento pelo serviço por meio do smartphone. Tudo isso, sem a necessidade de um contato direto com os proprietários. Além disso, por meio da IoT é possível que as empresas proprietárias desses equipamentos monitorem o estado de conservação e a localização dessas bicicletas e patinetes em tempo real. Assim, a integração entre os sistemas de comunicação e transporte favorecem a construção de espaços urbanos cada vez mais sustentáveis.

Entre os elementos que constituem as cidades inteligentes, temos o conceito de condomínios e edifícios inteligentes. A construção desses elementos é muito importante, pois representam o local onde reside ou trabalha boa parte da população urbana, principalmente nas metrópoles e grandes cidades do mundo. Ainda, mais de 40% de toda a energia consumida no mundo é atribuída aos edifícios comerciais e residenciais (KORKAS *et al.*, 2015). Todo esse dispêndio de energia ocorre devido à grande utilização dos sistemas de Aquecimento, Ventilação e Ar Condicionado (AVAC), iluminação, equipamentos de tecnologia da informação (computadores, impressoras e periféricos) e dispositivos de “plug-load” como celulares e tablets (WENG; AGARWAL, 2012). Assim, adotar estratégias que visem promover o bem-estar dos trabalhadores e moradores, reduzir os custos de manutenção e tornar esses espaços mais eficientes é fundamental para a construção de centros urbanos sustentáveis.

A utilização dos conceitos da Internet das Coisas auxilia no desenvolvimento de edifícios e condomínios inteligentes. Nas décadas de 1970 e 1980, o conceito de edifícios inteligentes estava relacionado ao consumo eficiente de energia e à possibilidade de realizar o controle de alguns equipamentos existentes nos edifícios através do computador pessoal (MORVAJ *et al.*, 2011). Atualmente, esse conceito engloba também a utilização de subsistemas capazes de realizar o gerenciamento do consumo de energia, de eletrodomésticos, de eletrônicos, dos dispositivos de AVAC e de sistemas de segurança (MORVAJ *et al.*, 2011). Portanto, através da utilização desses subsistemas é possível construir edifícios mais eficientes, utilizáveis, convenientes e lucrativos (DORSCH, 2018).

Com a utilização dos conceitos da IoT, diferentes aplicações existentes nos edifícios e nos condomínios inteligentes podem realizar uma troca de mensagens de maneira automática. Essas trocas podem ocorrer, por exemplo, entre o ar condicionado de um auditório e a central de climatização. A central de climatização pode analisar o calendário de eventos do auditório e regular automaticamente a temperatura desse ambiente a fim de preparar o local para a chegada dos participantes do evento. Assim, essa comunicação entre diferentes aplicações e edifícios gera a possibilidade de maior conforto e monitoramento constante de toda a energia consumida.

Por meio desse monitoramento e da ação de atuadores, os sistemas de iluminação, o AVAC e os demais equipamentos podem ter a utilização ajustada segundo a demanda de serviço. Desse modo, o consumo de energia é realizado de maneira eficiente, o que reduz custos e promove o desenvolvimento sustentável dos edifícios.

Assimile

Sustentabilidade, atualmente, é um conceito bastante utilizado. Ele indica as ações conscientes que possibilitam a manutenção das condições energéticas, informacionais e físico-químicas responsáveis pela continuidade de toda as formas de vida na terra, preservando o ambiente para as gerações futuras (BOFF, 2012). Assim, por meio da sustentabilidade todas as tecnologias empregadas visam promover o bem-estar dos seres vivos do planeta e assegurar o bem-estar da comunidade.

Os edifícios inteligentes são constituídos por sensores, atuadores, controladores, unidade central, interface, rede e medidores inteligentes (MORVAJ *et al.*, 2011). Cada uma dessas unidades possui um papel definido na construção de um edifício inteligente.

Os sensores são as unidades responsáveis por realizar o monitoramento das variáveis. Eles podem, por exemplo, medir a temperatura, o consumo de energia e identificar pessoas. Os atuadores têm a tarefa de interagir com o ambiente, e alguns exemplos de tarefas realizadas pelos eles são o envio de notificações de invasores, através de alarmes, e o ajuste da temperatura ambiente. Os controladores realizam o controle automático do ambiente através das regras estabelecidas pelos administradores do sistema. Esse controle pode ser utilizado, por exemplo, para manter a temperatura de um ambiente constante.

A unidade central é a responsável por permitir a programação e o gerenciamento de todo o sistema. É a unidade central, por exemplo, que recebe os dados dos diferentes sensores e garante a troca eficiente de mensagens de configuração e gerenciamento do sistema. As interfaces são utilizadas para interação entre os usuários e os sistemas. É por meio das interfaces que os usuários podem visualizar os dados obtidos e realizar o gerenciamento do edifício.

A rede representa a infraestrutura de comunicação entre todos os elementos constituintes do edifício inteligente. É através da infraestrutura que é possível estabelecer a comunicação entre os elementos de um edifício inteligente. Essa comunicação pode ser realizada tanto via cabos quanto por meio da comunicação sem fio. Os medidores inteligentes correspondem a unidades que podem realizar e apresentar as medidas localmente ou remotamente, podem agir como gateway para comunicação entre as diferentes redes existentes e permitem maior interatividade entre os usuários e a aplicação.

Todas essas unidades realizam as tarefas de maneira coordenada a fim de assegurar o aproveitamento eficiente de energia e melhorar o conforto dos usuários. A Figura 4.4 mostra algumas aplicações que podem ser realizadas a partir da utilização desses elementos.

Figura 4.4 | Exemplos de aplicações para a construção de edifícios inteligentes



Fonte: elaborada pelo autor.

Exemplificando

Um exemplo de edifício inteligente é o The Edge, localizado na cidade de Amsterdam, Holanda. Esse edifício tem sistemas inteligentes e mais de 28 mil sensores capazes de realizar o monitoramento constante de energia, auxiliar no descarte eficiente de lixo e realizar a integração dos espaços de estacionamento, convivência e trabalho, além de robôs para a limpeza do ambiente (**RANDALL, 2015**).

Sugerimos que assista ao vídeo indicado a seguir, disponível no YouTube, para conhecer mais as características desse edifício inteligente:

WORLD'S Greenest Office Building Is Dutch: The Edge. Amsterdam: Bloomberg Business, 2015. 1 vídeo (4 min 36 s). Publicado pelo canal Bloomberg.

Como grande parte do consumo de energia no planeta é realizada pelos edifícios e condomínios residenciais ou empresariais (KORKAS *et al.*, 2015), construir sistemas capazes de prover informações que gerem um consumo racional e eficiente de energia é fundamental para a construção das redes elétricas inteligentes (do inglês *smart grid*). Redes elétricas inteligentes correspondem à utilização de sensores, comunicação, inteligência computacional e controle na modernização dos sistemas de geração, transmissão e distribuição de energia (GELLINGS, 2009). Essas redes têm por objetivo realizar o monitoramento, a proteção e a otimização automática da operação e a manutenção dos elementos do sistema elétrico (GREER *et al.*, 2014). Nas redes elétricas inteligentes, existe um fluxo de informação e energia tanto da fonte emissora para os clientes do sistema quanto dos clientes para os fornecedores. Essa troca de informação, em conjunto com sistemas de análise e controle, permite a construção de redes elétricas mais eficientes.

A partir da utilização das redes elétricas inteligentes, os edifícios inteligentes podem, por exemplo, receber informações sobre a elevação do preço ou sobre a redução da confiabilidade no fornecimento de energia e, automaticamente, podem ajustar o consumo de energia para adequá-la aos parâmetros de fornecimento. Adicionalmente, os fornecedores de energia podem atribuir descontos aos consumidores para cada KWh reduzido no consumo. Assim, a união dos edifícios inteligentes e redes elétricas inteligentes tem um papel fundamental na utilização eficiente de energia.

Para utilizar o potencial transformador desses sistemas inteligentes é necessário que os dados possam ser transformados em informações que gerem valor de negócio para as organizações e para a sociedade (MANYIKA *et al.*, 2015). A utilização do pré-processamento local dos dados coletados por meio da computação em neblina auxilia na geração de valor para as aplicações IoT. Isso ocorre, pois, em vez de toda a tarefa de análise ser realizada pelas plataformas em nuvem, pode ser extraído conhecimento pelos dispositivos que estão mais próximos do usuário.

Reflita

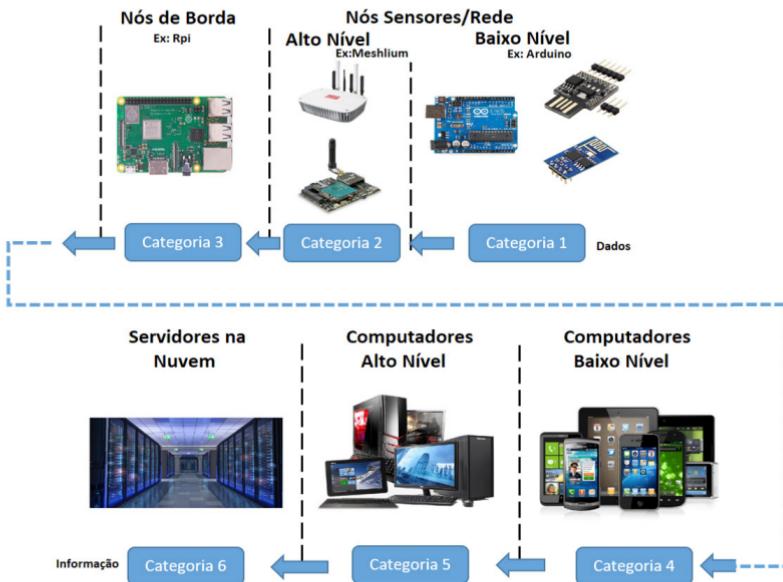
Foi dito que, no conceito da computação de borda, os dispositivos que geram os dados devem realizar um processamento local dessa coleta de informação. Entretanto, sabemos que os dispositivos da IoT normalmente têm disponibilidade reduzida de recursos. Quais os requisitos devem ser assegurados para garantir a utilização da computação de borda e em neblina para a IoT?

A computação em neblina define que os nós (dispositivos) da IoT podem realizar o processamento e o armazenamento dos dados antes de enviá-los para as plataformas em nuvem (YANNUZZI *et al.*, 2014). Para realizar esse processamento local é necessária uma maior capacidade de processamento do dispositivo. Assim, para manter as características pervasivas da IoT é necessário que a rede IoT seja composta de dispositivos com diferentes capacidades de processamento. Dessa forma, garante-se que cada uma das etapas de coleta, integração, processamento e análise de dados seja realizada de maneira mais eficiente, o que faz com que as características dos dispositivos da IoT e a funcionalidade da rede possam ser preservadas.

A Figura 4.5 mostra como os dados em uma aplicação IoT fluem através dos diferentes elementos da rede. Os nós sensores são os primeiros elementos da rede. Após coletados, os dados são enviados para os nós de borda, seguem para os computadores ou dispositivos processadores de baixo nível (smartphones e tablets), passam pelos sistemas computacionais mais avançados (computadores pessoais e notebooks) e chegam aos grandes servidores em nuvem. À medida que fazem o trajeto de coleta até os servidores em nuvem, esses dados são transformados em informação. Na Figura 4.5 é possível ver que a cada classe de elementos em que os dados são transferidos existe um aumento da capacidade de processamento (PERERA, 2017).

Por exemplo, a primeira etapa executada por uma aplicação IoT consiste em realizar a coleta de dados do ambiente. A classe de dispositivos responsável por coletar esses dados apresenta capacidade de processamento muito inferior ao realizado pelos servidores das plataformas em nuvem, que representam a última etapa de transferência de dados na IoT. Conforme novos equipamentos e sistemas são desenvolvidos, a capacidade de processamento e armazenamento de cada uma das classes é aumentada. Entretanto, a diferença de capacidade de processamento entre cada uma das classes permanece constante (PERERA, 2017). Por exemplo, a diferença do poder de processamento entre os dispositivos sensores e as plataformas em nuvem tende a permanecer constante, mesmo que ambas sofram um incremento. Isso ocorre, pois existem limitações impostas pelo processo de fabricação e pelo preço dos dispositivos. Para aumentar o poder de processamento, por exemplo, é necessário que também seja aumentado o tamanho do dispositivo (PERERA, 2017).

Figura 4.5 | Fluxo de dados entre os dispositivos em redes IoT com computação em neblina



Fonte: adaptada de Perera (2017).

Pesquise mais

No artigo a seguir, são apresentados alguns estudos de caso sobre diferentes aplicações que utilizam a computação em neblina. Dentre eles, são mostradas aplicações na agricultura inteligente, no transporte inteligente e no gerenciamento inteligente do lixo e da água. Esses estudos de caso são apresentados e analisados no contexto da computação em neblina e dos impactos positivos gerados nas cidades inteligentes.

PERERA, C. et al. Fog computing for sustainable smart cities: a survey. *ACM Computing Surveys (CSUR)*, v. 50, n. 3, p. 32, 2017.

Como pode ser visto, a união entre a computação em nuvem e os conceitos de computação em neblina contribui para o desenvolvimento e o amadurecimento das aplicações da Internet das Coisas. Por meio dessas tecnologias é possível aproveitar o potencial gerado pela IoT e proporcionar melhorias que afetem positivamente toda a sociedade.

Bons estudos!

Sem medo de errar

Grande parte do poder transformador da Internet das Coisas está na possibilidade de integrar diferentes dispositivos e sistemas para gerar o aprimoramento de processos que possam melhorar a vida das pessoas. Nesse sentido, o desenvolvimento de sistemas que auxiliem no acompanhamento remoto de pacientes proporciona maior comodidade e segurança para pessoas com algum tipo de enfermidade. A partir da geração e da análise de dados, a IoT possibilita, por exemplo, a melhoria da qualidade de vida de idosos e portadores de doenças crônicas.

Você foi contratado por uma grande empresa hospitalar para desenvolver um sistema de monitoramento remoto de paciente. Esse sistema deve possibilitar o monitoramento de pacientes por meio da leitura de vários sensores conectados ao corpo do paciente a fim de acompanhar os sinais vitais e outros sensores presentes na residência. Esses sensores têm o objetivo de coletar dados que possibilitem o acompanhamento, em tempo real, das condições de saúde do indivíduo. A partir de uma arquitetura que engloba os conceitos de computação em neblina, esses dados devem ser enviados para uma central e, quando alguma anomalia é identificada, devem ser tomadas medidas que preservem o bem-estar do paciente. Essas medidas podem ser, por exemplo, indicar tratamentos médicos ou acionar o serviço de resgate. Assim, esse sistema de monitoramento deve estar integrado ao serviço médico de resgate, pois a comunicação eficiente entre eles é crucial para garantir um atendimento rápido e eficaz ao paciente, especialmente nos serviços de urgência.

O Quadro 4.1 apresenta alguns dos sensores, atuadores e SBC que podem ser empregados para a construção desse sistema remoto de monitoramento de pacientes. Nesse quadro existem apenas exemplos de alguns elementos. Dependendo dos tipos de serviços oferecidos (monitoramento do sono, de atividades físicas ou do comportamento) outros dispositivos podem estar presentes. Para esse exemplo de resposta, como dispositivo responsável por realizar o tratamento inicial dos dados e gateway entre a rede dos sensores e a central de monitoramento na nuvem, foi utilizado o smartphone. A Figura 4.6 mostra um exemplo de conexão entre a rede dos dispositivos, a central de monitoramento e os serviços de resgate.

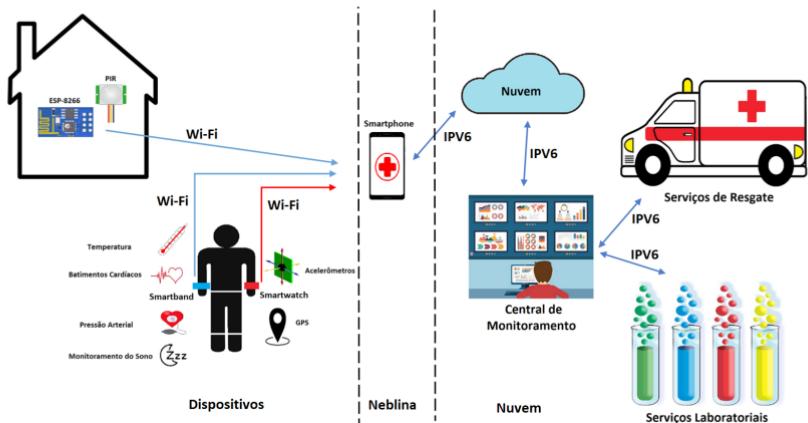
Quadro 4.1 | Exemplos de alguns elementos que podem constituir o sistema de monitoramento remoto de pacientes

Dispositivo	Sensor/atuador conectado	Função
Smartband	Monitor de frequência cardíaca, monitor de pressão arterial, monitor do sono e termômetro.	Realizar o monitoramento das funções vitais do paciente.
Smartwatch	GPS e pedômetro.	

Sensor PIR	Sensor de presença em cada cômodo.	Identificar o local na residência onde está o paciente.
ESP8266	X	Conexão do sensor PIR com o smartphone via Wi-Fi.
Smartphone	Podem ser utilizados o GPS, acelerômetros ou outros sensores, como elementos de redundância da rede, ou seja, sensores em duplicidade para melhorar a precisão e a robustez do sistema.	Conectar os dispositivos de monitoramento, realizar o primeiro processamento dos dados e comunicar com a internet.

Fonte: elaborado pelo autor.

Figura 4.6 | Exemplo de conexão entre os diferentes elementos para o sistema de monitoramento remoto de pacientes



Fonte: elaborada pelo autor.

Avançando na prática

Automação residencial utilizando os conceitos de computação em neblina

Você decidiu realizar a automação de sua residência. Para isso, você decidiu utilizar os conceitos de domótica aliados à computação em neblina e à computação de borda, pois deseja realizar um monitoramento em tempo real dos sistemas AVAC, de segurança e identificação de incêndios. A primeira etapa nesse processo corresponde à definição dos sensores, atuadores e SBC

que devem estar presentes para a realização dessa aplicação. Assim, você deve escrever um relatório inicial que contenha o levantamento de sensores, atuadores e SBC, além de elaborar um esquema que demonstre como esses diferentes dispositivos devem ser conectados utilizando os conceitos de computação em neblina.

Resolução da situação-problema

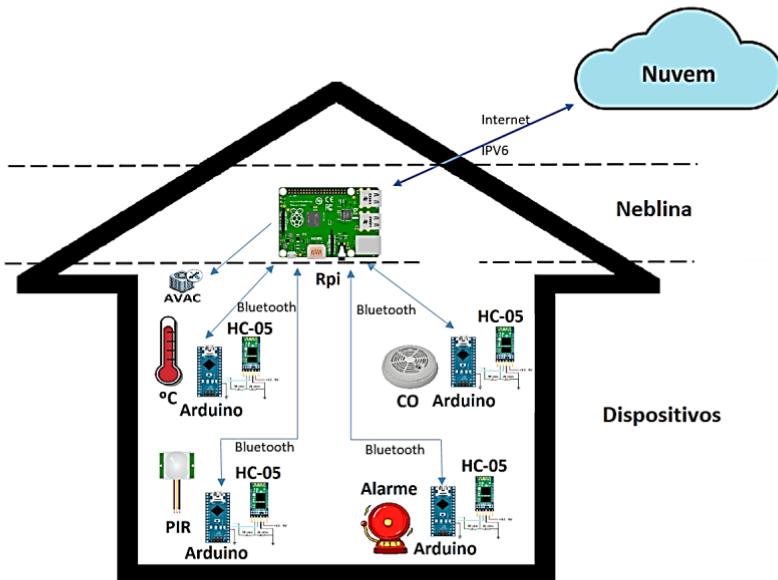
O Quadro 4.2 mostra alguns dos sensores, atuadores e SBC que podem ser utilizados para construir o sistema de domótica utilizando os conceitos de computação em neblina. Para esse sistema é sugerida a adoção do Raspberry Pi como elemento responsável por realizar a comunicação entre os diferentes sensores/atuadores e realizar o processamento inicial dos dados. Nesse cenário, o Rpi passa ser utilizado como gateway e nó de processamento. A Figura 4.7 exibe um esquema de conexão entre os diferentes elementos dessa solução.

Quadro 4.2 | Sugestão de sensores, atuadores e SBC utilizados para a construção da aplicação em domótica

Dispositivo	Sensor/atuador conectado	Função
Sensor PIR	PIR	Identificar a presença de pessoas nos cômodos da residência.
Alarme sonoro	Alarme	Responsável por emitir sinais sonoros quando identificar algum invasor na residência ou identificar fumaça.
Sensor de temperatura	DS18B20	Responsável por coletar dados de temperatura ambiente.
Termostato	Atuador sobre os elementos AVAC	Manter a temperatura do ambiente por meio da regulação da potência do ar-condicionado.
Sensor de CO	MQ-7	Medir as concentrações de CO2 no ambiente para identificação de incêndios.
Arduino nano	X	Conectar os sensores/atuadores a fim de realizar a comunicação com o Raspberry Pi.
HC-05	X	Placa conectada ao arduino que permite a comunicação via Bluetooth como o Rpi.
Raspberry Pi	X	Realizar o processamento inicial dos dados e possibilitar a conexão com a internet.

Fonte: elaborado pelo autor.

Figura 4.7 | Exemplo de esquema para conexão dos diferentes elementos utilizados na aplicação de domótica



Fonte: elaborado pelo autor.

Uma outra possível solução seria a substituição do arduino+HC-05 pela placa ESP8266 e realizar a comunicação com o Rpi por meio do protocolo Wi-Fi ao invés do Bluetooth.

Faça valer a pena

1. A união entre os dispositivos da IoT e as plataformas de computação em nuvem representa um grande passo no sentido de tornar a Internet das Coisas uma tecnologia cada vez mais eficiente e versátil. Entretanto, existem alguns requisitos, especialmente para aplicações que demandam monitoramento e ações em tempo real, que não são completamente atendidos por essa união.

Para solucionar esses problemas não contemplados pela união com a computação em nuvem e tornar a IoT mais robusta, foi desenvolvido o paradigma de computação em neblina.

Dado o contexto apresentado, marque a alternativa que contém a definição correta do paradigma computação em neblina.

- É um paradigma que mostra como os dados devem ser coletados e compartilhados entre cada um dos dispositivos presentes na rede.

Ele tem por finalidade definir os protocolos a serem utilizados para a troca de mensagens entre os dispositivos dentro da rede local.

- b. É um paradigma que proporciona o envio de dados para a internet sem nenhum elemento de tratamento primário. Assim, todos os dispositivos da rede podem comunicar e enviar dados diretamente com as plataformas em nuvem.
- c. É um paradigma que define uma camada ou um dispositivo intermediário que realiza um processamento primário local antes de enviar os dados brutos para as plataformas em nuvem. Tem por finalidade reduzir o número de mensagens trocadas entre os dispositivos e a nuvem.
- d. É um paradigma que define como as mensagens entre os gateways e os dispositivos devem ser trocadas. Tem por finalidade não permitir que os dispositivos realizem comunicação direta entre si, o que reduz a quantidade de mensagens trocadas na rede.
- e. É um paradigma que mostra a forma como os dispositivos e demais elementos da rede devem ser conectados. Assim, ele define uma nova arquitetura de rede para a IoT, o que proporciona a redução de dispositivos na rede e aumenta a área de abrangência da rede.

2. O desenvolvimento de aplicações IoT e a união com várias outras tecnologias como a computação em nuvem contribuíram para a difusão da Internet das Coisas para, praticamente, todas as áreas do conhecimento humano. A conexão de áreas como transporte, logística, serviços de saúde, educação, governo e infraestrutura por meio da IoT e redes de comunicação, possibilitou o surgimento das cidades inteligentes. A seguir, constam algumas afirmativas sobre as cidades inteligentes.

- I. São espaços urbanos que privilegiam o desenvolvimento econômico em detrimento da convivência e das relações pessoais.
- II. São espaços urbanos em que há integração e cooperação entre diferentes serviços, a fim de possibilitar um desenvolvimento sustentável.
- III. Nas cidades inteligentes, a união entre as tecnologias de comunicação e a IoT visa gerar informação para a melhoria dos serviços prestados e para o bem-estar da comunidade.
- IV. Nas cidades inteligentes, as relações pessoais são perdidas, pois todas as interações entre pessoas ocorrem apenas no meio virtual.

Considerando as afirmações apresentadas, marque a opção correta.

- a. Apenas as afirmativas I e IV estão corretas.
- b. Apenas as afirmativas II e III estão corretas.
- c. Apenas as afirmativas I, II e III estão corretas.
- d. Apenas as afirmativas II e IV estão corretas.
- e. Apenas as afirmativas II, III e IV estão corretas.

3. A integração entre os dispositivos da IoT, computação em neblina, de borda e em nuvem representa uma grande oportunidade para a construção de aplicações cada vez mais robustas e eficientes. A seguir, são apresentadas algumas afirmativas sobre os conceitos de computação em neblina, de borda e em nuvem.

- I. A computação em nuvem possibilita que os dispositivos da IoT utilizem recursos, virtualmente infinitos, de servidores remotos.
- II. A computação em neblina visa reduzir a latência e a largura de banda necessária para a transmissão de mensagens na IoT.
- III. A computação de borda corresponde ao processamento realizado no próprio nó responsável por gerar dados.
- IV. Para a aplicação dos paradigmas de computação em neblina e de borda não é necessário que os dispositivos tenham maior capacidade de processamento.

A partir do exposto, marque a alternativa que contém apenas afirmativas corretas.

- a. Apenas as afirmativas I e II estão corretas.
- b. Apenas as afirmativas II e IV estão corretas.
- c. Apenas as afirmativas II e III estão corretas.
- d. Apenas as afirmativas II, III e IV estão corretas.
- e. Apenas as afirmativas I, II e III estão corretas.

Seção 2

Veículos conectados

Diálogo aberto

A Internet das Coisas (IoT), em conjunto com outras tecnologias, possibilitou o desenvolvimento de aplicações que só eram criadas pelos efeitos especiais do cinema. Grandes diretores utilizam da imaginação para criar, virtualmente, aplicações que se tornaram realidade graças à IoT. Carros autônomos, casas e cidades conectadas, robôs que desempenham serviços domésticos e realizam cirurgias médicas, além de assistentes pessoais cibernetícios são exemplos de cenas de filmes que, atualmente, estão presentes no mundo real.

Dentre as várias ideias futurísticas presentes no cinema, os carros conectados e autônomos têm um papel de destaque. Não é difícil encontrar produções que vislumbram a utilização de veículos autônomos em um sistema de transporte completamente integrado em que veículos, pedestres e infraestrutura urbana funcionam de maneira coordenada. Grande parte das tecnologias ilustradas nesses filmes tornaram-se realidade graças ao desenvolvimento da Internet das Coisas. Vários sistemas como transporte, moradia, educação e saúde empregam as tecnologias presentes na IoT para promoverem maior segurança, conforto e comodidade aos usuários.

Nesta seção, você está na segunda etapa de construção do sistema de monitoramento de pacientes. Nela, você deve indicar como ocorre a integração entre os serviços de emergência e resgate com a central de monitoramento de pacientes, destacando o papel dos veículos conectados. Os veículos conectados, para essa aplicação, são representados pelas unidades de resgate. Portanto, são de extrema importância, pois são esses serviços os responsáveis por realizarem o atendimento dos pacientes nos momentos mais críticos. Assim, é necessário que esse relatório contenha um esquema que represente a conexão e os protocolos empregados para a integração com a central de monitoramento. Além disso, esse esquema deve conter como os veículos de resgate devem estar integrados aos sistemas de comunicação a fim de possibilitar que eles sejam contatados assim que uma situação de emergência for identificada nos dados do paciente monitorado.

Para ajudá-lo nessa tarefa, serão estudados os conceitos básicos sobre veículos conectados e alguns sensores e atuadores utilizados tanto pelos veículos quanto para o gerenciamento do tráfego. Assim, será possível

empregar esses conceitos para a construção dessa etapa do projeto de monitoramento de pacientes.

Agora você já tem várias ferramentas necessárias para criar aplicações para a resolução de problemas reais. Não deixe passar essa oportunidade de fazer a diferença na vida das pessoas. Coloque em prática tudo o que você aprendeu e participe da construção de um mundo melhor. Vamos começar?

Não pode faltar

O desenvolvimento de cidades inteligentes está intimamente relacionado à criação e ao amadurecimento dos sistemas inteligentes de transporte (do inglês Intelligent Transportation System – ITS). Os sistemas de transporte inteligente aplicam as tecnologias da Internet das Coisas em todos os meios de transporte a fim criar um ambiente integrado e proporcionar, a partir da análise de dados, um melhor planejamento e gerenciamento dos sistemas de mobilidade urbana (MUTHURAMALINGAM *et al.*, 2019). Por meio do ITS é possível proporcionar a redução da poluição e melhorar a segurança, o conforto e a eficiência dos serviços de transporte. Tudo isso é alcançado por meio da análise e da transmissão, em tempo real, da informação. Essa maior eficiência do sistema de transporte é gerada pela integração de sensores e ações de atuadores localizados em vários setores das cidades inteligentes como vias de tráfego, estacionamentos e veículos conectados.

Na construção e efetiva aplicação do ITS, são necessárias a integração e a cooperação de diferentes áreas que compõem as cidades inteligentes. Sistemas como carros conectados, estacionamentos inteligentes, infraestrutura, sinalização automatizada e transporte público inteligente são exemplos de áreas que compõem o ITS (ALAM *et al.*, 2016). Nesse cenário, em papel de destaque, estão os veículos conectados (LU *et al.*, 2014).

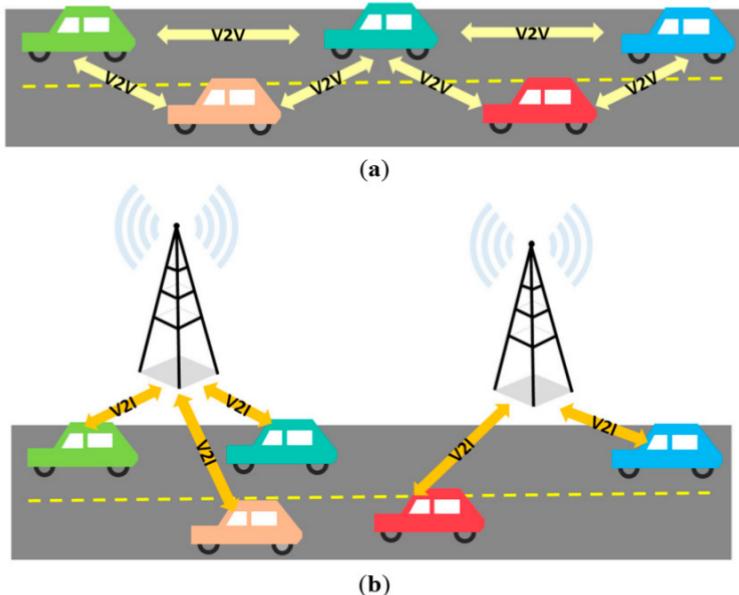
Segundo Uhlemann (2015), um veículo conectado corresponde a aplicações, serviços e tecnologias que o conectam ao ambiente ao redor por meio da internet. Para isso, vários dispositivos são adicionados ao veículo a fim de possibilitar a conexão entre diferentes sistemas dentro dele ou com redes, aplicações e serviços ao redor. Essa comunicação pode ocorrer através de redes entre os próprios veículos (V2V, ou vehicle-to-vehicle) ou entre os veículos e a infraestrutura (V2I, ou vehicle-to-infrastructure). A Figura 4.8 apresenta uma representação desses tipos de comunicação. Na comunicação V2V, as mensagens são trocadas entre os próprios veículos. Por exemplo, por meio dessa comunicação é possível que veículos que estiverem envolvidos em um acidente possam transmitir mensagens de alerta para outros veículos próximos. A comunicação V2V é derivada do padrão IEEE 802.11. Quando

dois ou mais veículos estão em uma mesma área de cobertura, eles podem, automaticamente, estabelecer uma rede de comunicação. Desse modo, é possível que cada veículo atue como um roteador da rede criada, permitindo que as mensagens sejam trocadas entre o maior número possível de veículos (KHAIRNAR *et al.*, 2014).

Através da comunicação V2I, ao entrar em uma área de cobertura, o veículo que recebeu esse alerta pode transmitir essa mensagem para uma central de monitoramento informando o local e o horário do acidente. Essa comunicação V2I corresponde à troca de mensagens entre os veículos e a infraestrutura de transporte, que ocorre, normalmente, através das Unidades de Transmissão (UT). As UT, além de receberem essas mensagens, também podem transmitir, por exemplo, informações sobre as condições de trânsito e comunicar rotas alternativas de locomoção. A Figura 4.9 mostra uma ilustração dessa aplicação.

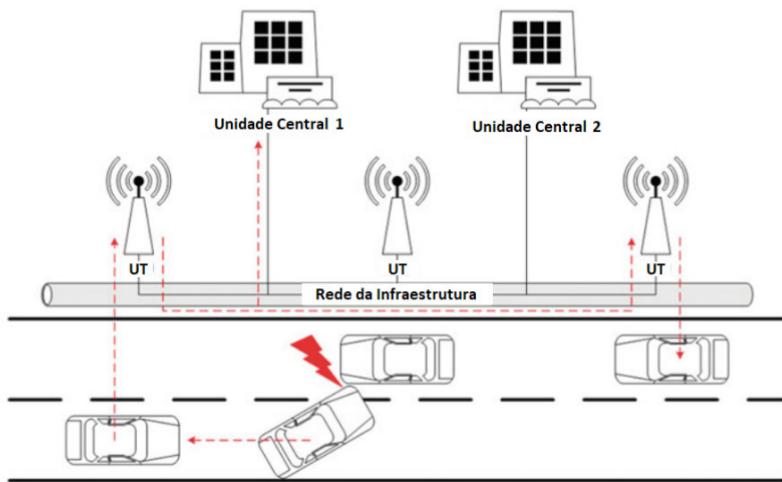
Em comunicações de curta distância pode ser utilizado, por exemplo, o Radio-frequency identification (RFID) (ARENA; PAU, 2019). Esse tipo de comunicação V2I pode ser empregado para tarefas de pagamento automático, detecção de vagas em estacionamentos e identificação de veículos.

Figura 4.8 | Ilustração das comunicações V2V e V2I: (a) representação da comunicação V2V; (b) representação da comunicação V2I



Fonte: Parrado e Donoso (2015).

Figura 4.9 | Ilustração de aplicação envolvendo redes V2V e V2I



Fonte: adaptada de Alam *et al.* (2016).

Essa interação cooperativa entre os diferentes elementos que compõem o sistema de transporte inteligente, a fim de melhorar a segurança, a sustentabilidade, a eficiência e o conforto por meio da comunicação e da cooperação, é chamada de Transporte Inteligente Cooperativo (C-ITS, do inglês Cooperative Intelligent Transportation System) (ALAM *et al.*, 2016). Essa cooperação é alcançada por meio da troca de mensagens através das redes sem fio V2V e V2I.

Para a construção dos sistemas cooperativos de transporte é necessário que exista a comunicação eficiente entre os integrantes de toda a estrutura de transporte. A fim de possibilitar essa interação eficiente são utilizados, basicamente, dois tipos de comunicação (UHLEMANN, 2015). A comunicação de curta distância, por meio da tecnologia IEEE 802.11p, e as redes celulares, através das tecnologias 4G e 5G. O padrão IEEE 802.11p define os requisitos das camadas físicas e de enlace para permitir a comunicação em curtas distâncias e de alta velocidade entre veículos conectados (MIUCIC, 2018). Por meio do IEEE 802.11p é possível estabelecer uma comunicação direta entre os veículos ou entre veículos e as estações bases do ITS. Essa rede de comunicação móvel entre carros ou entre carros e a infraestrutura é chamada de redes VANET (Veicular Ad Hoc Networks) (ALVES *et al.*, 2009). A comunicação por meio das redes 4G e 5G corresponde à utilização de tecnologias móveis de transmissão de dados que permite alta velocidade

de transmissão e baixas latências por meio de redes celulares (DAHLMAN *et al.*, 2013).

Não apenas os carros conectados utilizam a IoT para tornar o trânsito mais seguro. As rodovias, ruas, avenidas e demais vias de tráfego também empregam os sensores e atuadores para auxiliar no gerenciamento e na melhoria do transporte. Especialmente nas grandes cidades, a utilização da IoT auxilia na redução dos congestionamentos. Com o aumento do número de carros nas ruas, os congestionamentos passam a ser grandes responsáveis pelo aumento da poluição sonora e atmosférica, pelo aumento considerável do tempo de viagem e pelo desperdício de combustível (BULL, 2003). Todas essas características fazem com que o sistema de mobilidade urbana seja completamente afetado pelos quilômetros de congestionamentos existentes nas grandes cidades. Sendo assim, a IoT passa a ser uma ferramenta importante para auxiliar na redução dos congestionamentos e no aproveitamento inteligente da malha viária.

Por meio da utilização de análise de vídeos, dos sistemas adaptativos de controle de tráfego (ATCS, do inglês Adaptive Traffic Control System) e redes de sensores é possível melhorar o gerenciamento dos sistemas de transporte e reduzir o número de quilômetros de congestionamento (SONI; SARASWAT, 2017). Através de câmeras e sensores localizados nas principais vias de tráfego, imagens das condições do trânsito são capturadas e transmitidas para uma estação central. De posse desses vídeos, é possível obter dados estatísticos sobre o comportamento, em tempo real, das vias monitoradas. Dados como velocidade média dos veículos, quantidade de veículos, acidentes e infrações podem ser gerados através desse método.



Acesse o QR Code para consultar uma aplicação que utiliza a análise de vídeos e auxilia os motoristas a encontrarem vagas de estacionamento.

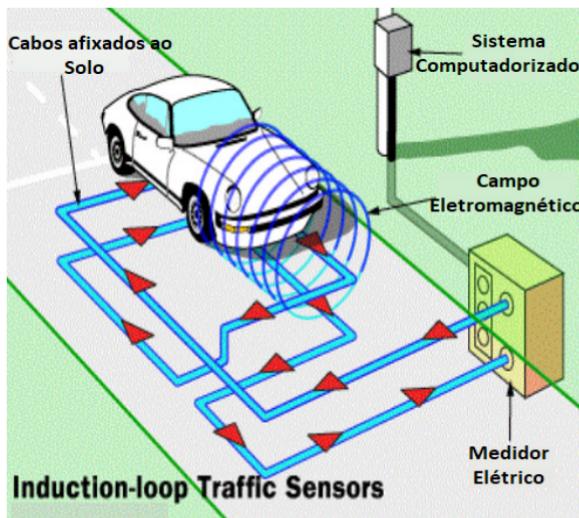
<https://bit.ly/2n8L9J0>

Os sistemas adaptativos de controle de tráfego são uma estratégia de gerenciamento de tráfego em que os tempos dos semáforos são ajustados, em tempo real, a partir da análise das condições atuais das vias, da capacidade do sistema e da previsão de demanda (STEVANOVIC, 2010). Ao ajustar os tempos dos semáforos é possível ajustar o fluxo de veículos de diferentes vias de tráfego. Esse ajuste possibilita que regiões críticas de uma cidade possam ter um fluxo mais homogêneo de veículos, o que reduz os tempos das viagens e os congestionamentos.

Existem vários sensores que auxiliam na redução dos congestionamentos das grandes cidades e no gerenciamento de todo o sistema de transporte.

Sensores como detectores por laço de indução (detectores de loop), células de carga, sensores infravermelhos e de identificação por radiofrequência colaboram para a melhoria dos sistemas de transporte. Os detectores de loop funcionam através do princípio de indução eletromagnética (SONI, 2017). Laços magnéticos são posicionados no solo e quando algum veículo passa por esse laço é gerada uma corrente elétrica que é detectada por uma central. Esse tipo de sensor pode ser utilizado para identificar a presença ou para realizar a contagem de veículos. A Figura 4.10 ilustra o princípio de funcionamento desse sensor. Esses detectores por laço de indução podem ser utilizados, por exemplo, para realizar a contagem de veículos que estejam trafegando em cada uma das faixas de uma via. Além disso, esses detectores também são empregados para identificar infrações de trânsito como invasão de faixa exclusiva e avanço de sinal vermelho.

Figura 4.10 | Princípio de funcionamento do sensor detector de loop



Fonte: adaptada de Somashekhar, Shirabadagi e Hegadi (2014).

As células de carga são transdutores eletromecânicos que medem a deformação ou flexão de um determinado corpo e convertem essa deformação em valores de corrente ou de tensão (FARUOLO; MARTHA, 2008). A deformação sofrida é proporcional ao peso do corpo que provocou essa deformação. Assim, é possível aferir o peso de veículos e estimar a quantidade de pessoas que utilizam o transporte público (LETSHWITI; LAMPRECHT, 2004).

Os sensores de infravermelho também são bastante utilizados em todo o sistema de transporte. Esses sensores contêm um emissor e um receptor

que são responsáveis por identificar a presença de outros carros, pessoas e obstáculos e por medir a velocidade de veículos (GUERRERO-IBÁÑEZ; ZEADALLY; CONTRERAS-CASTILLO, 2018). Eles podem ser empregados, por exemplo, na abertura automática de cancelas e portões, além de serem utilizados para a contagem de veículos que trafegam em uma estrada (HAYAMA; MINAKATA, 2008).

Os identificadores por radiofrequência (RFID, do inglês Radio-frequency identification) são empregados para realizar o rastreamento de veículos e de carga (PANDIT *et al.*, 2009), a detecção de congestionamentos (AL-NAIMA; HAMD, 2012) e para prover serviços de estacionamento e pedágio de maneira automática (PALA; INANC, 2007).

Pesquise mais

Sugerimos a leitura do artigo a seguir, que mostra a importância dos sensores em todo o sistema de transporte e o papel deles na melhoria da segurança e do gerenciamento do trânsito:

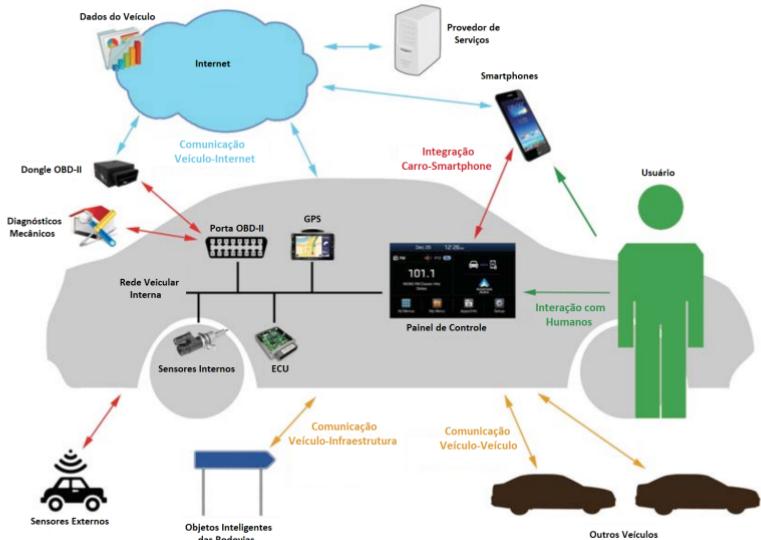
GUERRERO-IBÁÑEZ, J.; ZEADALLY, S.; CONTRERAS-CASTILLO, J. Sensor technologies for intelligent transportation systems. *Sensors*, [s.l.], v. 18, n. 4, p. 1.212, 2018.

Devido à característica pervasiva da IoT, existem vários sensores presentes nos veículos. Eles são os grandes responsáveis por melhorar a segurança em todo o sistema de transporte, além de possibilitar o surgimento dos veículos conectados e autônomos (COPPOLA; MORISIO, 2016). Segundo Greengard (2015), atualmente, existem mais de 70 unidades de controle eletrônico (ECU, do inglês Electronic Control Units) em apenas um veículo, mais de 100 MB de códigos binários e milhões de linhas de códigos construídas para gerar veículos cada vez mais seguros e eficientes. Por exemplo, são encontradas ECU responsáveis por gerenciamento dos sistemas de navegação, comunicação, entretenimento, emissão de gases, suspensão, pressão dos pneus, iluminação, piloto automático e prevenção de colisão. Para realizar a comunicação entre todas essas centrais e sensores são necessários protocolos confiáveis e até 4 km de cabos (QUANTOS..., 2017).

Todos esses cabos são responsáveis por conectar vários sistemas diferentes e por possibilitar comunicação em alta velocidade. Sistemas como de arrefecimento, alimentação, injeção eletrônica, freios, ignição e de entretenimento devem estar interligados para que seja possível realizar as tarefas de maneira coordenada. Devido à grande quantidade de sensores, atuadores e sistemas de comunicação, os automóveis conectados passam a desempenhar o papel

de nós móveis para o ITS (COPPOLA; MORISIO, 2016). Informações como condições do trânsito, relato de acidentes e indicações de melhores rotas podem ser compartilhadas entre os veículos conectados. Nesse sentido, cada veículo conectado passa a receber e a transmitir informações, em tempo real, a fim de tornar o trânsito mais seguro e eficiente. A Figura 4.11 mostra alguns dos sistemas presentes nos veículos conectados.

Figura 4.11 | Exemplo de sistemas presentes nos veículos conectados



Fonte: adaptada de Coppola e Morisio (2016).

Na Figura 4.11, os sensores internos (embarcados) do veículo são responsáveis por monitorar as variáveis de funcionamento de todo o automóvel. Variáveis como temperatura do óleo, nível de carga da bateria e qualidade da mistura ar-combustível são captadas e enviadas para as ECU. Na ECU, esses dados são analisados e podem ser enviados, automaticamente, para as centrais de diagnóstico, por meio da conexão com a internet ou do acesso local por meio da porta OBD-II. De posse desses dados é possível realizar um diagnóstico preciso das condições do veículo e executar possíveis manutenções. Em seus smartphones os usuários podem consultar os dados históricos do automóvel e agendar revisões diretamente com a concessionária. Por meio da rede de comunicação estabelecida entre os veículos e a infraestrutura é possível que informações sobre incidentes ou alterações nas condições da estrada possam ser compartilhadas de maneira automática entre todos os veículos pertencentes a essa rede. Desse modo, os sistemas embarcados

e a troca de informações contribuem para a construção de um sistema de trânsito cada vez mais seguro.

Através dos sistemas embarcados, os veículos conectados podem oferecer diferentes serviços aos motoristas. Eles podem ser de segurança no trânsito, de entretenimento, de eficiência no tráfego, de redução nos custos de rodagem, além de conveniência e interação (EVERIS, 2015). Os serviços de assistência no trânsito auxiliam o motorista na prevenção de acidentes e falhas mecânicas. Por exemplo, sensores e câmeras podem ser instalados dentro do veículo para monitorar o estado de atenção do motorista. Com eles é possível verificar os sinais biológicos, o movimento e o intervalo de tempo em que os olhos permanecem fechados. Por meio desses sinais é possível identificar se o motorista está cansado. Caso confirmem que o estado de atenção está comprometido, podem ser enviados sinais de alerta e mensagens para que o condutor descanse (GHIMIRE et al. 2015). Outros sistemas utilizados para evitar acidentes consistem em utilizar radares, laser e câmeras para identificar a iminência de um acidente. Assim, podem auxiliar o motorista com a frenagem automática ou preparar o carro para uma colisão, por exemplo, tensionando os cintos de segurança e ligando para os serviços de emergência (SIVAKUMAR; MANGALAM, 2014).

Refletá

Durante esta seção foram apresentados os sistemas embarcados e os diferentes sensores que constituem os veículos conectados e as possibilidades geradas pela conexão entre esses veículos, a infraestrutura de transporte e a internet. Como todo sistema de comunicação, existem questões relativas à segurança que devem ser observadas. Por exemplo, como garantir que as informações de localização do veículo só possam ser acessadas por pessoas autorizadas. Sendo assim, você conseguiria elencar algumas das ameaças à segurança da informação que podem acometer os veículos conectados?

Os serviços de entretenimento e comodidade fornecem maior conforto aos usuários por meio da central integrada de mídias. Podem ser oferecidos serviços de multimídia integrados a smartphones, a redes internas de comunicação Wi-Fi e ao compartilhamento de músicas e vídeos. Através da comunicação entre os veículos, as redes sociais podem ser utilizadas para que os ocupantes compartilhem com outros motoristas as informações da viagem, para que peçam apoio na estrada ou para trocarem experiências (LUAN et al., 2015). Sensores inteligentes de temperatura também podem identificar padrões dos usuários e ajustar automaticamente a temperatura, conforme as preferências de cada passageiro.

Nos sistemas para a eficiência no trânsito são utilizados GPS e redes de telefonia para fornecer uma localização precisa e em tempo real dos veículos. A partir dessa localização, podem ser enviadas informações, por exemplo, sobre preços de combustíveis, pontos turísticos ou vagas de estacionamento que estejam nas proximidades. Além disso, por meio dos sistemas de comunicação com outros veículos (V2V) e com a infraestrutura de trânsito (V2I) é possível que os veículos conectados compartilhem informações sobre as condições das vias e que auxiliem na indicação de melhores rotas e na redução de congestionamentos (COPPOLA; MORISIO, 2016).

Para a redução dos custos de rodagem, pode ser utilizada a telemetria do veículo. Por exemplo, por meio de dispositivos conectados à porta On-Board Diagnostics (OBD) podem ser enviados, diretamente para as agências de seguro, dados sobre o comportamento do condutor. O OBD é um sistema que proporciona acesso e diagnóstico de várias ECU do veículo. Com ele é possível monitorar dados como a velocidade do veículo, o nível e o consumo de combustível, a emissão de poluentes, o odômetro, a posição dos pedais, e ler possíveis códigos de erro existentes nas ECU do veículo. A partir de todos esses dados é possível que as seguradoras mapeiem o comportamento do motorista. Assim, podem ajustar os serviços oferecidos, conforme as demandas de cada tipo de condutor (EVERIS, 2015).

Assimile

Apesar das similaridades entre os veículos conectados e veículos autônomos, eles apresentam definições diferentes. Nos veículos conectados existe a integração e comunicação entre sistemas por meio da internet. Em um veículo autônomo também existe essa integração entre diferentes sistemas e comunicação via internet. A diferença entre essas duas tecnologias reside no fato de que para os veículos conectados existe a necessidade de um condutor, já em veículos autônomos não é necessário a existência de um motorista (COPPOLA; MORISIO, 2016).

Para a maior conveniência dos motoristas e ocupantes, pode existir, por exemplo, a conexão entre os veículos e as casas conectadas. Essa conexão permite que os motoristas consigam realizar tarefas como abrir o porão da garagem, controlar luzes, verificar níveis de combustível e ligar remotamente o veículo (FORD, 2016). Além disso, por meio da telemetria do veículo é possível identificar falhas mecânicas ou momentos de manutenção preventiva e, automaticamente, realizar o agendamento de serviços de reparo.



Acesse o QR Code para consultar uma aplicação que demonstra a utilização de algoritmos de visão computacional a fim de identificar as faixas de trânsito em uma rodovia. Sistemas similares a essa aplicação são empregados para possibilitar a construção de veículos autônomos.

<https://bit.ly/2najWWc>

Essas são apenas algumas das possibilidades geradas pelos veículos conectados. A partir delas é possível perceber que o desenvolvimento desses veículos é fundamental para a construção de sistemas de transporte mais eficientes. Portanto, os veículos conectados contribuem para a construção de um trânsito mais seguro e para a preservação do meio ambiente.

Sem medo de errar

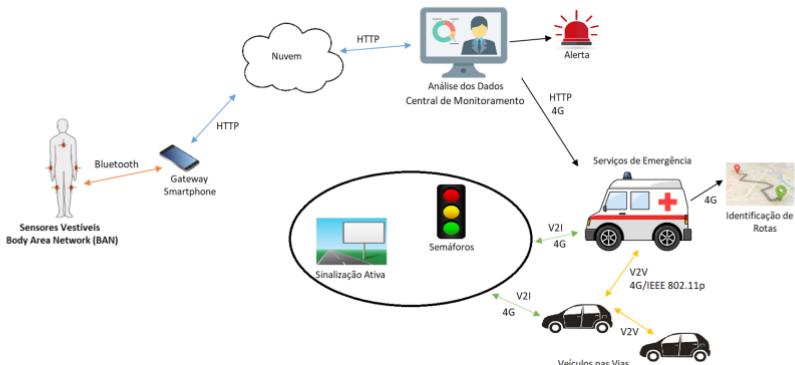
Os veículos conectados têm um papel fundamental para o desenvolvimento do sistema de transporte inteligente. Como esses veículos passam a ser vistos como nós móveis do ITS, é através deles que informações sobre o trânsito são compartilhadas. Esse compartilhamento de informação pode ocorrer, por exemplo, entre os veículos presentes nas vias ou entre os veículos e os demais elementos de infraestrutura como semáforos e painéis eletrônicos de mensagens. Nesse sentido, os veículos conectados podem integrar e beneficiar outros sistemas que compõem as cidades inteligentes, por exemplo, aos serviços inteligentes de emergência e de cuidado com à saúde.

Você está trabalhando no desenvolvimento de um sistema inteligente de monitoramento de pacientes. Na primeira etapa, você iniciou a construção de um relatório que deve ser apresentado aos diretores de uma grande rede hospitalar. Agora você está na segunda etapa de desenvolvimento desse relatório. Nela, deve ser apresentado um esquema que contém os dispositivos necessários para a integração do sistema inteligente de monitoramento com os serviços de resgate de emergência. Além disso, devem ser indicados os protocolos que permitem a integração entre os diferentes elementos que compõem essa aplicação. Assim, será possível apresentar um esquema que resume os dispositivos utilizados e indica como deve ocorrer a integração entre eles.

Para que o sistema de monitoramento de pacientes proporcione maior qualidade de vida aos pacientes assistidos, especialmente em situações de emergência, é necessário que esse ele seja integrado aos serviços de resgate de emergência. Isso ocorre, pois são esses serviços os responsáveis por prestar o primeiro atendimento aos pacientes em casos de urgência. Assim, reduzir o tempo necessário para esses serviços se deslocarem até o local onde está o paciente é fundamental para a preservação da saúde desse indivíduo.

Uma sugestão de esquema de integração entre os serviços de monitoramento e resgate de emergência é expresso na Figura 4.12. Nessa figura é possível ver alguns protocolos empregados para a comunicação entre os diferentes dispositivos e unidades que compõem o sistema proposto.

Figura 4.12 | Exemplo de esquema para o sistema de monitoramento de pacientes



Fonte: elaborada pelo autor.

Os dispositivos “vestíveis” instalados no paciente monitorado formam uma rede de comunicação responsável por coletar os seus dados vitais. Essa rede tem o nome de Body Area Network (BAN). A comunicação entre os elementos dessa rede e um smartphone (gateway) pode, por exemplo, ocorrer através do protocolo Bluetooth. Através do smartphone, esses dados podem ser enviados, por meio do protocolo Hypertext Transfer Protocol (HTTP), para plataformas em nuvem. Com a integração desses dados na nuvem, as centrais médicas de monitoramento podem realizar uma análise e, caso algum comportamento anormal seja identificado, é gerado um alerta. Esse alerta pode ser enviado diretamente para os serviços de resgate. Ao receber esse alerta e as informações sobre a localização, além de alguns dados, como tipo sanguíneo, alergias e estado atual de saúde do paciente, os serviços de resgate passam a utilizar os carros conectados e o ITS para reduzir o tempo para realizar o atendimento de urgência.

De posse dos dados de localização do paciente, a central de resgate pode encontrar a unidade móvel mais próximo do local da emergência. Essa comunicação pode ser realizada, por exemplo, através das redes 4G. Assim que a unidade móvel é contatada, ela pode estabelecer a comunicação com outros veículos conectados através do V2V, indicando que está em um trabalho de resgate e que necessita de caminho livre para realizar esse procedimento. Logo, os motoristas podem ser avisados sobre a rota da unidade

de resgate mesmo antes de obter um contato visual com essa unidade. Desse modo, podem liberar a passagem e tornar o trajeto mais rápido. Essa comunicação também pode ocorrer entre os veículos e as infraestruturas de trânsito (V2I), como semáforos e painéis de informação, a fim de mostrar aos demais veículos que não estão integrados a essa rede que eles estão na rota de uma unidade em trabalho de resgate.

Esses são apenas alguns exemplos de integração entre os diferentes dispositivos e sistemas responsáveis pelo monitoramento do paciente em situações de emergência. Portanto, por meio dessa solução proposta, você está contribuindo para a melhoria da qualidade de vida dos pacientes monitorados. Além disso, com a implementação desse sistema, é possível que várias vidas sejam salvas por conta da integração de diferentes sistemas que compõem as cidades inteligentes.

Avançando na prática

Sistema automático para detecção de colisão e acionamento de serviços médicos

Uma grande empresa de seguros deseja oferecer aos clientes um serviço automático de detecção de colisão e acionamento de resgate. Devido ao seu conhecimento sobre carros conectados, você foi convidado para desenvolver um sistema capaz de identificar um acidente e, automaticamente, informar às autoridades policiais e aos serviços de emergência a localização desse sinistro.

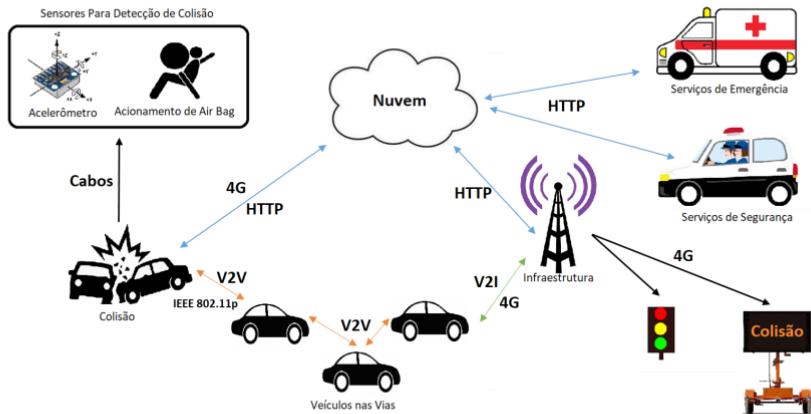
Para a apresentação desse projeto aos diretores da empresa, você deve mostrar um esquema que contenha alguns dos dispositivos e protocolos a serem utilizados na construção dessa aplicação. Esse esquema deve possibilitar uma visão geral da integração necessária entre os veículos conectados e os serviços policiais e de emergência.

Resolução da situação-problema

Para a detecção automática de colisão podem ser utilizados acelerômetros e sensores que constatam o acionamento do airbag. Após a identificação da colisão são enviadas mensagens para os demais veículos que estão trafegando na via, a fim de indicar o local da colisão e prevenir que novos acidentes ocorram em decorrência dessa colisão. Essas mensagens podem ser trocadas entre os veículos através da comunicação V2V, utilizando o protocolo IEEE 802.11p. Ao identificar essa colisão, por exemplo, dados sobre o local e

danos causados aos veículos são enviados para as plataformas em nuvem e são encaminhados para os serviços de emergência e autoridades policiais por meio da internet, utilizando o protocolo HTTP. Além disso, esses dados podem ser enviados às unidades de monitoramento que utilizam a infraestrutura do trânsito para informar aos motoristas, através da comunicação V2I e de redes celulares 4G, sobre o acidente e facilitar o resgate das vítimas. A Figura 4.13 apresenta um esquema de conexão dos vários sistemas constituintes dessa aplicação.

Figura 4.13 | Exemplo de esquema para a identificação de colisão e acionamento de serviços médicos



Fonte: elaborada pelo autor.

Faça valer a pena

1. Os veículos conectados já são uma realidade. Por meio da conexão com a internet e com sistemas que permitem a comunicação com outros veículos e a infraestrutura, esses veículos podem contribuir para uma maior segurança no trânsito e para a preservação do ambiente.

Assinale a alternativa que contém a afirmativa correta sobre os veículos conectados.

- Os veículos conectados realizam a comunicação apenas via internet. Isso ocorre, pois esse tipo de conexão exige altas taxas de transmissão e grande banda de dados.
- Os veículos conectados podem interagir com outros veículos por meio da comunicação V2V ou com a infraestrutura, por meio da comunicação V2I.

- c. A comunicação entre os veículos conectados e a infraestrutura de trânsito só pode ocorrer via redes celulares, pois essa é a única forma de comunicação com a internet existente nas rodovias.
- d. Para a comunicação entre veículos conectados, normalmente é utilizada a tecnologia Bluetooth, pois ela apresenta altas taxas de transmissão e baixo consumo de energia.
- e. A integração entre os carros conectados e o ITS só pode ocorrer em vias urbanas, pois exige a capacidade de conexão com a internet.

2. Os veículos conectados têm vários sensores, atuadores e sistemas de comunicação que auxiliam na construção de um trânsito mais seguro e com maior comodidade aos condutores. Por meio dos veículos conectados, podem ser oferecidos alguns serviços aos motoristas. Entre eles, podemos citar os serviços que auxiliam na segurança no trânsito, na conveniência e na interatividade e na redução dos custos de rodagem. A seguir são feitas algumas afirmações sobre esses serviços.

- I. Os serviços de segurança no trânsito são realizados, exclusivamente, pelos sensores internos dos veículos conectados.
- II. Os serviços de segurança no trânsito podem auxiliar os motoristas por meio de sistemas ativos como Anti-lock Braking System (ABS) e passivos como airbag.
- III. Os serviços de interatividade proporcionam interação entre os usuários do mesmo veículo e outros veículos conectados que estejam nas vias.
- IV. Os custos de rodagem podem ser reduzidos por meio de serviços de telemetria e acompanhamento de manutenções.

Assinale a alternativa que contém as afirmativas corretas sobre os serviços oferecidos aos usuários de veículos conectados.

- a. I e II, apenas.
- b. II e IV, apenas.
- c. I, II e III, apenas.
- d. II, III e IV, apenas.
- e. I, III e IV, apenas.

3. Os veículos conectados auxiliam na construção de um trânsito mais seguro. À medida que assistem os motoristas para uma condução mais segura e são utilizados como nós do ITS, os veículos conectados passam a desempenhar um papel fundamental na construção das cidades inteligentes. A seguir são feitas algumas afirmativas sobre o papel dos veículos conectados na construção de cidades inteligentes.

- I. Os veículos conectados agem como nós do ITS, pois podem receber, enviar e transmitir mensagens por meio das redes VANET.
- II. O que distingue um veículo conectado de um veículo autônomo é a quantidade de atuadores que existe em cada uma das ECUs.
- III. Os veículos conectados contribuem para a redução dos congestionamentos por meio do compartilhamento de informação entre veículos e infraestruturas de trânsito.
- IV. As comunicações V2V e V2I permitem o ajuste automático de sinais de trânsito e o controle do fluxo de veículos nas vias.

Assinale a alternativa que contém as afirmativas corretas a respeito dos veículos conectados.

- a. I e II, apenas.
- b. II e III, apenas.
- c. II, III e IV, apenas.
- d. I, II e IV, apenas.
- e. I, III e IV, apenas.

Seção 3

IoT na prática

Diálogo aberto

Vários problemas enfrentados por nossa sociedade ainda não foram completamente solucionados. Problemas ambientais gerados pelo consumo desordenado de recursos ambientais, poluição, desperdícios, além da escassez de alimentos e acesso universal à saúde são questões que afligem, principalmente, os países em desenvolvimento. Adotar estratégias que visem ao aproveitamento racional e eficiente de recursos, reduzir desperdícios e proporcionar o acesso global a serviços essenciais são ações fundamentais para o desenvolvimento de nossa sociedade. Para alcançar esses objetivos, um fator primordial é a informação. Com informação é possível conhecer os processos, melhorar o planejamento e o gerenciamento, identificar falhas, propor melhorias, aumentar a produtividade, reduzir custos e democratizar o acesso a serviços.

Atualmente, uma das tecnologias responsáveis por gerar dados, analisá-los e prover informações para a tomada de decisão é a Internet das Coisas (IoT). Diferentes áreas do conhecimento humano, setores da sociedade e economia fazem uso de aplicações da IoT para compreender o ambiente em que estão inseridas, por meio da análise de dados e da geração de conhecimento. Portanto, a IoT surge como uma possível ferramenta para ajudar na construção de uma sociedade mais justa e eficiente, em que o desenvolvimento econômico possa ser alcançado por meio de aplicações sustentáveis que promovam o crescimento e que preservam o ambiente e as relações sociais.

Retomando nossa situação-problema, você está trabalhando em uma aplicação que gera maior acesso aos serviços de saúde por meio do acompanhamento remoto de pacientes. Nesta última etapa, você deve finalizar o relatório a ser entregue aos diretores da grande rede hospitalar. Para isso, nesse relatório deve ser acrescentado um modelo de negócios para essa aplicação. Esse modelo de negócios deve possibilitar uma visão geral das características dessa aplicação e uma avaliação da viabilidade econômica do projeto. Assim, esse modelo de negócios deve definir o valor do negócio, o público-alvo, os clientes, o método de produção e as fontes de despesas e receitas.

Para auxiliá-lo nessa tarefa, nesta unidade serão apresentadas aplicações de IoT no setor industrial, agrícola e na melhoria dos serviços de saúde. Além disso, também será mostrado como deve ser construído um modelo de negócio para o desenvolvimento e gerenciamento de aplicações IoT. Assim,

você estará preparado para desenvolver soluções IoT que contribuem para a construção de um mundo melhor.

Não pode faltar

A Internet das Coisas já é realidade em nossas vidas. Você já parou para pensar que a IoT já está em nosso cotidiano e que a presença dela nem é mais sentida? Quando utilizamos, por exemplo, os aplicativos de navegação e compartilhamos informações sobre as condições de trânsito e essas informações são exibidas, automaticamente, pelos painéis eletrônicos das estradas, estamos utilizando a presença ubíqua da IoT. Quando empregamos os smartwatches, smartbands e smartphones para pagar contas em maquininhas de cartões de crédito, também utilizamos a comodidade gerada pela IoT. Essas e outras aplicações da IoT que utilizamos diariamente e nem percebemos a presença dessa tecnologia, mostram que a Internet das Coisas realizou transformações profundas em nossa sociedade (WEISER, 1991).

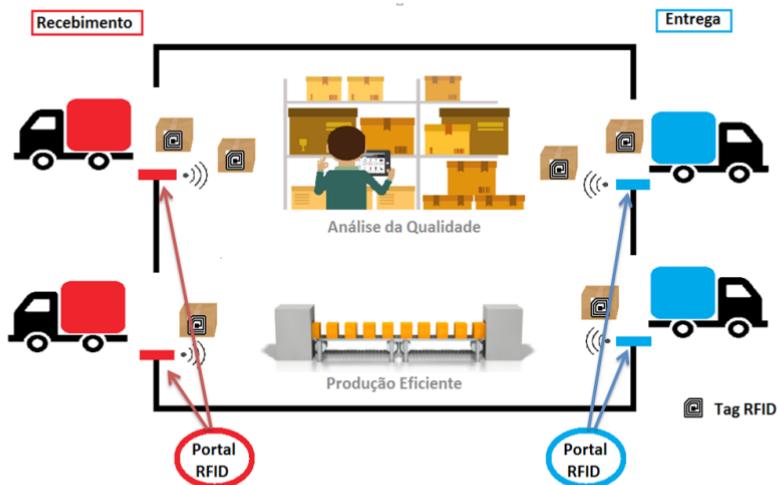
A IoT é empregada em várias áreas, tais como em atividades industriais, logística, monitoramento de processos e manufatura (BORGIA, 2014). O projeto EURIDICE é um exemplo de aplicação IoT, desenvolvido pela União Europeia, que pretende criar um sistema integrado de logística (SCHUMACHER; GSCHWEIDL; RIEDER, 2010). Esse projeto visa desenvolver uma plataforma centralizada de informações sobre cada tipo de item transportado no território europeu e as interações existentes entre esses itens, as pessoas e os diferentes sistemas de transporte inteligente. Para isso, são utilizados chips passivos de Radio-Frequency Identification (RFID). Esses chips devem ser adicionados a cada item transportado no território europeu. A partir da união desses chips com os sistemas de infraestrutura de comunicação é possível rastrear esses objetos e definir o tempo, o local e o status de cada item presente em uma carga. Assim, é possível que cada objeto seja rastreado e que as informações obtidas sejam compartilhadas por diferentes áreas, como o sistema de transporte e a indústria produtora. Esse compartilhamento de informações possibilita que todos os elementos envolvidos no sistema produtivo possam prover serviços específicos e customizados para que seja alcançada maior eficiência na distribuição de cargas.

A RFID e a rede IoT também são utilizadas para gerar um maior número de informação, aplicar análise de dados e obter maior valor de negócio. O sistema proposto por Cai *et al.* (2014), utiliza o RFID e as redes de sensores sem fio para conseguir informações durante todo o ciclo de vida de um produto. Tags passivos de RFID podem ser afixados em cada um dos produtos. Quando esses produtos passam, por exemplo, por leitores de RFID, é possível obter informações precisas sobre esse item. Informações como

composição e destinatário podem ser armazenadas e distribuídas entre as diferentes áreas da empresa.

A Figura 4.14 mostra um exemplo de aplicação. Nessa figura, cada matéria-prima recebida é automaticamente identificada através da RFID. Durante o processo produtivo, todas as partes constituintes de um produto podem ser corretamente identificadas e catalogadas. Além disso, cada item pode ser fabricado segundo as características desejadas por cada cliente. Desse modo, é possível que exista uma integração entre a linha de produção e o sistema de logística. Assim, os componentes necessários para o processo produtivo, transporte, armazenamento e venda ao consumidor final podem ser rastreados pelo sistema proposto. Seja por questões legais, para assegurar a qualidade dos produtos ou garantir um melhor gerenciamento da produção, a rastreabilidade é essencial para a eficiência do sistema produtivo (DA SILVA, 2004). Nesse sentido, a integração de diferentes sensores e a capacidade de comunicação ubíqua da IoT contribuem para a criação de indústrias cada vez mais dinâmicas e competitivas.

Figura 4.14 | Exemplo de utilização do RFID na indústria

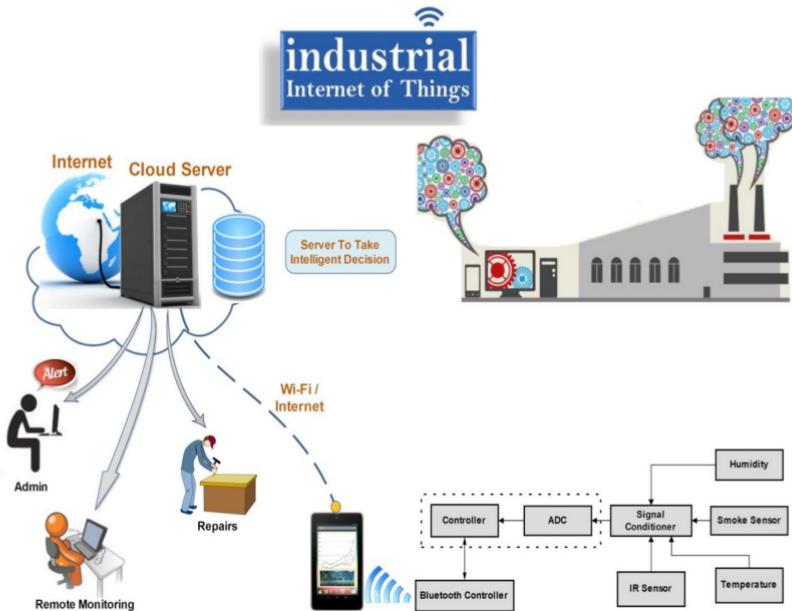


Fonte: elaborada pelo autor.

O monitoramento de processos industriais também emprega a IoT para um melhor aproveitamento de recursos. A indústria automobilística, por exemplo, emprega a IoT para possibilitar o diagnóstico em tempo real de algum problema que possa acometer algum veículo (HANK *et al.*, 2013). Por exemplo, os sensores de desgaste das pastilhas de freio podem identificar e informar quando a capacidade de frenagem do veículo está comprometida.

devido ao desgaste excessivo das pastilhas de freios. Sistemas de monitoramento automático do processo produtivo também utilizam a IoT. Deshpande, Pitale e Sanap (2016) sugerem um modelo de monitoramento em tempo real do ambiente industrial por meio da IoT e da análise de dados gerada por algoritmos de inteligência artificial. A Figura 4.15 mostra uma visão geral do sistema sugerido.

Figura 4.15 | Visão geral do sistema de monitoramento industrial



Fonte: adaptada de Deshpande, Pitale e Sanap (2016).

No sistema presente na Figura 4.15, são adicionados sensores para monitorar temperatura, pressão, umidade, vibrações e detecção de invasores. Esses sensores geram dados para que seja possível compreender o estado atual do ambiente industrial. Os dados são enviados para um sistema Android por meio de conexões Bluetooth. Quando é identificado que alguma dessas variáveis ultrapassaram limites definidos previamente, são enviadas mensagens de alerta para os administradores do sistema. Os dados obtidos pelos sensores também são enviados por meio do protocolo IEEE 802.11 para servidores em nuvem, que empregam algoritmos de inteligência artificial para indicar as melhores soluções para cada problema encontrado. Por exemplo, podem ser utilizadas as redes neurais artificiais para identificar e corrigir falhas em redes de sensores sem fio (VIEIRA; ALMEIDA; MEIRELES, 2017). Assim, é

possível existir um monitoramento automático e em tempo real do ambiente produtivo, o que possibilita a identificação e a correção mais precisa de problemas que possam prejudicar a produção.



Acesse o QR Code para consultar uma aplicação que utiliza o Arduino e uma ponte H para realizar o controle de velocidade de um motor de corrente contínua. Esse processo é de grande importância para as indústrias, pois é por meio dos motores que as cargas são deslocadas e o trabalho é realizado.

<https://bit.ly/2lCDmCG>

Com a introdução e o desenvolvimento de novos sensores, atuadores e protocolos de comunicação, outras áreas também passam a utilizar a IoT para a construção de aplicações que auxiliem no processo produtivo. Na agricultura, por exemplo, vários processos são automatizados a fim de alcançarem maiores rendimentos e qualidade dos produtos, além de reduzir os impactos causados pelo uso excessivo dos recursos naturais (KAMILARIS *et al.*, 2016).

Refita

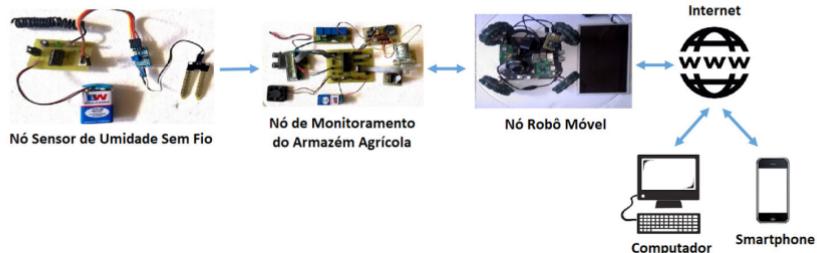
Grande parte das aplicações da Internet das Coisas utiliza as características pervasivas dessa tecnologia para monitorar diferentes processos e ambientes. Você saberia elencar algumas aplicações em que essa característica seja fundamental?

A agricultura é uma aplicação altamente imprevisível (KAMILARIS *et al.*, 2016). Isso ocorre, pois ela é dependente das condições climáticas (períodos de chuva e estiagem) e de outros eventos imprevisíveis, como doenças em animais e pragas. Desse modo, coletar informações que possibilitem aos administradores tomarem decisões mais precisas sobre as estratégias de gerenciamento da produção é de grande importância para o sucesso de uma cultura agrícola. Nesse sentido, a IoT pode ser utilizada como uma ferramenta valiosa para a coleta, a integração e a análise de dados que possam subsidiar a tomada de decisão.

A utilização dos recursos hídricos, por exemplo, representa um dos pilares para o desenvolvimento da agricultura (BERNARDO, 1997). Quando não existe o aproveitamento eficiente desse recurso podem ser gerados impactos ambientais, econômicos e sociais para as áreas que não utilizam os sistemas de irrigação de maneira sustentável (BERNARDO, 1997). Portanto, alguns sistemas que empregam as redes de sensores, atuadores e outros protocolos de comunicação foram desenvolvidos para proporcionar o aproveitamento racional desse recurso.

Um exemplo de sistema desenvolvido para o aproveitamento racional da água na agricultura é proposto por Gondchawar e Kawitkar (2016). Esse sistema consiste em três diferentes nós: um robô baseado em Global Positioning System (GPS), uma unidade inteligente de irrigação e uma unidade de gerenciamento do armazém agrícola. A função do robô é realizar detecção de umidade, espantar pássaros e outros animais, realizar a vigilância e a capina do ambiente agrícola. O nó de irrigação tem a função de capturar dados em tempo real das condições ambientais e realizar o controle da umidade do solo. No nó de gerenciamento do armazém agrícola existem sensores que captam a umidade do ar, temperatura e detectam a presença de invasores. Essas três unidades utilizam o protocolo ZigBee e a rede Wi-Fi para realizarem a troca de informação dentro da rede interna e possibilitarem a comunicação com redes externas pela internet. A Figura 4.16 mostra uma visão geral do sistema proposto. A partir da utilização desse sistema é possível ter um controle inteligente da irrigação e de outras atividades agrícolas. Assim, a produção agrícola gera menos impacto ambiental e produz alimentos com maior qualidade.

Figura 4.16 | Visão geral do sistema agrícola proposto por Gondchawar e Kawitkar (2016)



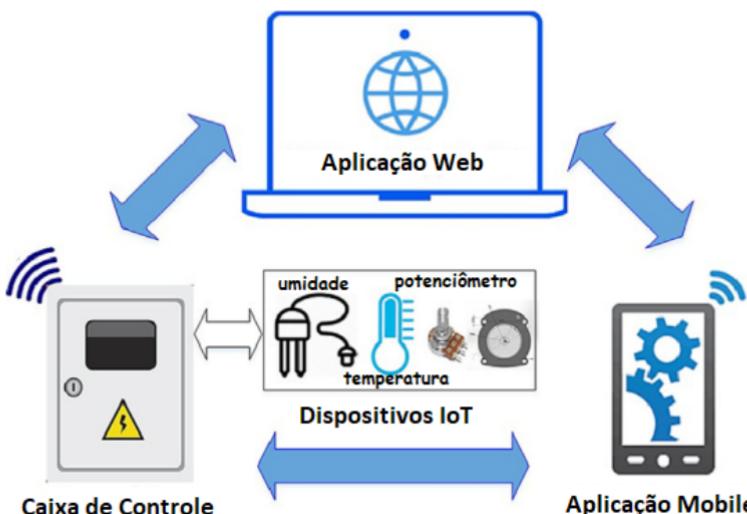
Fonte: adaptada de Gondchawar e Kawitkar (2016).

Outro sistema que utiliza hardware, aplicação web e aplicativo de celular para obter e analisar dados, a fim de gerar maior valor agregado ao sistema de produção agrícola, é proposto por Muangprathub *et al.* (2019). O hardware desenvolvido consiste em uma caixa de controle que é responsável por coletar dados sobre a plantação, por exemplo, sensores para medir a umidade do solo. Esses dados são enviados para uma aplicação web que tem a função de integrar e aplicar algoritmos de mineração de dados a fim de encontrar padrões que possam ser empregados no melhor aproveitamento da produção. A aplicação desenvolvida para o smartphone tem por finalidade permitir aos usuários que realizem o controle da produção e que recebam notificações sobre estado geral da plantação. A Figura 4.17 mostra uma visão geral do sistema proposto. Por meio desse sistema é possível monitorar grandes áreas plantadas, melhorar a qualidade e reduzir os custos de produção.



Acesse o QR Code para consultar o desenvolvimento de uma aplicação para monitoramento de um sistema agrícola. Nesse sistema, é utilizado o Arduino e um smartphone com sistema operacional Android para realizar o acompanhamento de algumas variáveis climáticas.
<https://bit.ly/2IFI8AD>

Figura 4.17 | Visão geral do sistema de monitoramento agrícola proposto por Muangprathub *et al.* (2019)



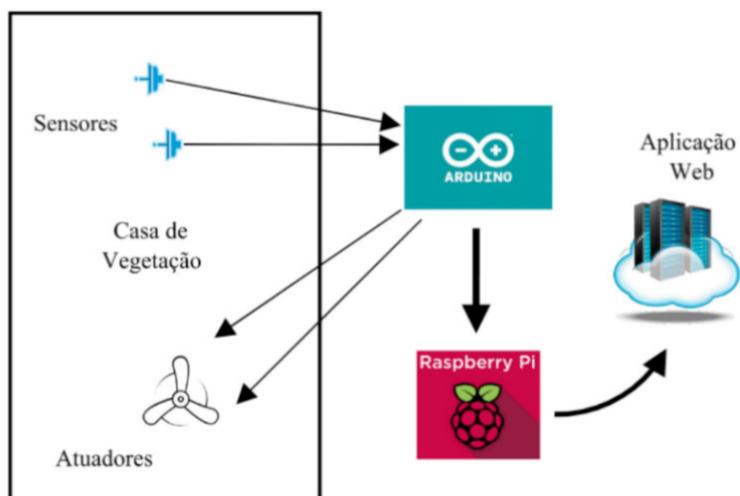
Fonte: adaptada de Muangprathub *et al.* (2019).

Na agricultura de precisão, especialmente em casas de vegetação, a IoT auxilia no monitoramento preciso da cultura para antecipar e aumentar a produção. A agricultura de precisão corresponde à aplicação da tecnologia para aumentar a eficiência do sistema agrícola com base no manejo diferenciado da cultura (TSCHIEDEL; FERREIRA, 2002). Nesse sentido, a agricultura de precisão consiste no gerenciamento agrícola a partir de informações precisas geradas por várias tecnologias empregadas no campo (ROZA, 2000).

As casas de vegetação correspondem ao sistema de cultivo em que são utilizadas estruturas com cobertura de material transparente para proteger as plantas da exposição direta aos fatores climáticos (BELTRÃO; FILHO; FIGUEIRÉDO, 2002). Em sistemas agrícolas como casas de vegetação,

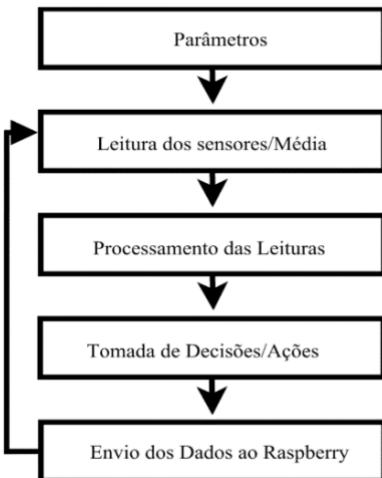
utilizar a IoT para implementar a agricultura de precisão gera um ambiente ideal para o cultivo de mudas. Um exemplo de aplicação da IoT para o cultivo de mudas é apresentado por Ferraz Filho (2018). No sistema proposto são empregadas placas de prototipagem Arduino e Raspberry Pi (Rpi), sensores (de temperatura, umidade e luminosidade) e atuadores (relés, exaustores, válvulas solenoides e aquecedores) integrados a uma aplicação web para realizar o monitoramento e o controle das condições climáticas da casa de vegetação. Essa aplicação foi empregada para propiciar o aumento e a antecipação do crescimento de mudas. A Figura 4.18 mostra a visão geral do sistema proposto. Os sensores e atuadores são conectados ao Arduino por meio da comunicação cabeada. A partir do Arduino os dados são enviados para o Rpi por meio da comunicação serial e, posteriormente, são enviados para a aplicação web. A Figura 4.19 apresenta o fluxograma para o software criado. Por meio dessa aplicação, os dados podem ser visualizados e, automaticamente, são acionados os atuadores a fim de manter os parâmetros (umidade, temperatura e luminosidade) do cultivo das mudas dentro de limites estabelecidos. Alguns exemplos de atuadores utilizados por esse sistema são lâmpadas incandescentes, válvulas solenoides, aquecedores e exaustores. Por meio desse sistema, por exemplo, quando a luminosidade não está adequada, podem ser acionadas as lâmpadas ou, à medida que a temperatura da estufa atinge um limite inferior estabelecido, são ligados os aquecedores.

Figura 4.18 | Visão geral do sistema de monitoramento de mudas em casas de vegetação proposto por Ferraz Filho (2018)



Fonte: adaptada de Ferraz Filho (2018).

Figura 4.19 | Fluxograma do software



Fonte: adaptada de Ferraz Filho (2018).

Exemplificando

Um exemplo de sistema que utiliza as características ubíquas da IoT para possibilitar o monitoramento de sistemas é o Smart Garden Hub. Nesse sistema é possível realizar o monitoramento de jardins pelo smartphone. Vários sensores são integrados em um sistema que possibilita, por exemplo, realizar o controle automático da umidade do solo e reduzir o consumo hídrico. Além disso, por meio do Smart Garden Hub podem ser gerados programas que permitem que o usuário defina padrões de preferência. O vídeo a seguir apresenta algumas das funcionalidades desse sistema:

GREENIQ Smart Garden Hub - Control Your Garden with Your Smartphone.
[S. l.: s. n.], 2018. 1 vídeo (2 min 12 s). Publicado pelo canal GreenIQ.

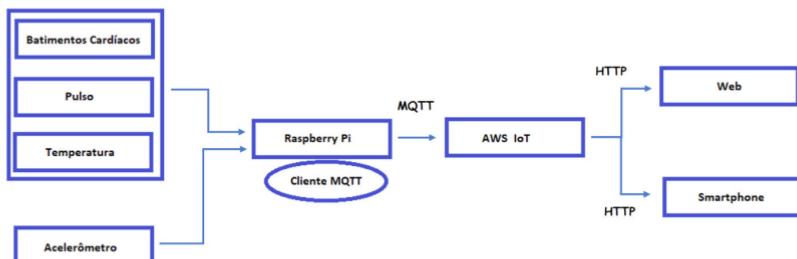
Uma das áreas de maior interesse para IoT corresponde à utilização das características pervasivas de seus componentes a fim de promover melhores cuidados médicos e monitoramento da saúde (PANG, 2013). Dentre as várias oportunidades geradas pela adoção da IoT, podem ser destacadas o monitoramento remoto de pacientes, o acompanhamento de programas de exercícios físicos, o auxílio a pacientes com doenças crônicas e o cuidado com idosos (ISLAM *et al.*, 2015). As aplicações da IoT na área médica visa promover a qualidade de vida dos pacientes assistidos, reduzir custos no

tratamento, universalizar o acesso a serviços de saúde e melhorar a experiência dos usuários desses serviços (ISLAM *et al.*, 2015).

Rani *et al.* (2017) propõem um sistema para monitoramento remoto de pacientes capaz de coletar vários sinais vitais deles, como batimentos cardíacos, intensidade das pulsações, nível de oxigênio sanguíneo e temperatura corporal. Esses dados podem ser acessados via web ou por aplicativos de smartphones. Além disso, ele é capaz de identificar alguma anomalia nesses dados e gerar alertas para o médico responsável. O sistema proposto é composto de sensores, de um Raspberry Pi (Rpi) e utiliza a ferramenta AWS IoT para a consolidação dos dados. Esses dados podem ser acessados pela web ou por aplicativos de smartphones. A Figura 4.20 mostra o diagrama de blocos para o sistema proposto.

Os sensores são conectados ao Rpi por meio de cabos. Através do protocolo MQTT, esses dados são enviados para a plataforma em nuvem AWS IoT. Nessa plataforma, os dados podem ser visualizados tanto pela web quanto por aplicativos de smartphones. Eventos como quedas dos pacientes geram alertas que são enviados diretamente para o médico responsável. Assim, é possível realizar o monitoramento remoto do paciente e receber notificações sobre comportamentos nos dados que indiquem problemas com os pacientes.

Figura 4.20 | Visão geral do sistema de monitoramento de pacientes proposto por Rani *et al.* (2017)

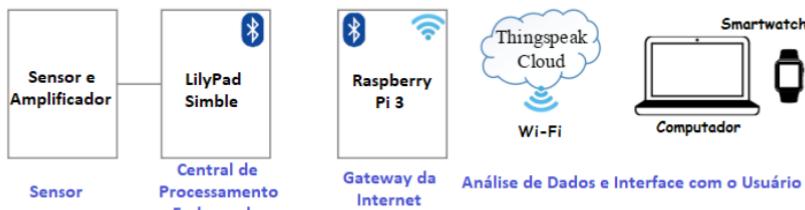


Fonte: adaptada de Rani *et al.* (2017).

Brueck *et al.* (2018) apresentam uma aplicação que utiliza a IoT para monitorar o grau de hidratação de praticantes de atividades físicas através do suor. Esse monitoramento ocorre em tempo real e pode ser visualizado por meio da internet. Para esse sistema foi desenvolvido um sensor capaz de medir o suor do atleta. A Figura 4.21 apresenta a arquitetura desse sistema. Esse sensor é conectado à variante do Arduino, conhecida como LilyPad. Após receber esses dados, do LilyPad eles são enviados através do protocolo Bluetooth para um Rpi, que os envia para a plataforma em nuvem Thingspeak. Dessa plataforma, eles podem ser analisados e acessados via web

ou por aplicativos. Assim, é possível realizar o monitoramento em tempo real do estado de hidratação do atleta.

Figura 4.21 | Visão geral do sistema de monitoramento de atividades físicas proposto por Brueck *et al.* (2018)



Fonte: adaptada de Brueck *et al.* (2018).

O monitoramento de pacientes portadores de doenças crônicas também utiliza a IoT para proporcionar maior qualidade de vida ao enfermo. Raji, Jeyasheeli e Jenitha (2016) desenvolveram um sistema capaz de realizar o acompanhamento remoto dos sinais vitais de pacientes com doenças crônicas. Para isso, são utilizados sensores de temperatura, de batimentos cardíacos e de pressão sanguínea. Os dados obtidos por esses sensores são integrados pelo Arduino. No próprio Arduino é realizado o processamento desses dados. Esse processamento consiste em aplicar algoritmos de aprendizado de máquina, como o Naive Bayes e o Support Vector Machine, para classificar o estado de saúde do paciente como normal ou anormal. Portanto, através desse sistema é possível obter a classificação do estado do paciente e tomar decisões baseadas nos dados obtidos.

Na assistência aos idosos também existem aplicações que utilizam a IoT para proporcionar maior independência e comodidade às pessoas da terceira idade. Um exemplo de sistema desenvolvido para esse fim é apresentado por Hua *et al.* (2018). Para realizar esse monitoramento, são utilizados sensores de temperatura, pulsação e acelerômetro. Esses sensores são conectados ao Single Board Computer (SBC) Intel Edison, que é responsável por integrar esses dados. Além desses sensores, existe uma pulseira que tem um botão de emergência. Quando esse botão de emergência é pressionado são enviadas mensagens de texto (SMS, do inglês Short Message Service) para pessoas cadastradas. Essa mensagem contém a senha para a abertura da porta onde o idoso se encontra. Todos os dados gerados são enviados para um webserver construído para essa aplicação e podem ser acessados via internet. Assim, por meio dessa aplicação é possível realizar um monitoramento remoto de idosos e garantir o acesso a esse paciente em situações de emergência.

Para colocar em prática essas ideias e construir um negócio que possa ser lucrativo e contribuir para o crescimento da sociedade, é necessário

desenvolver um plano de negócios para definir os objetivos da aplicação e quais passos devem ser dados para que esses objetivos sejam alcançados.

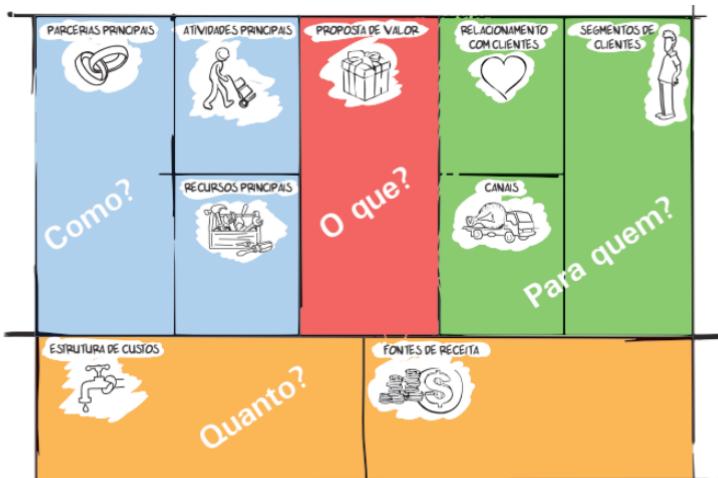
O plano de negócios pode ser visto como uma estratégia de planejamento do negócio focada nos objetivos da empresa ou no seu público-alvo (BRASIL *et al.*, 2016). Essa estratégia utiliza da criatividade, de boas ideias e de originalidade para planejar o negócio através da clara definição dos objetivos e das ações que devem ser implementadas para alcançar os objetivos traçados.

Pesquise mais

O portal do Serviço Brasileiro de Apoio às Micro e Pequenas Empresas (Sebrae) é um excelente local para aprender mais sobre como construir um negócio de forma mais precisa e eficiente. Através desse portal é possível realizar alguns pequenos cursos e ter acesso a diferentes materiais para a construção de uma empresa.

Uma ferramenta bastante interessante que auxilia no desenvolvimento do plano de negócio e no gerenciamento estratégico para um empreendimento é conhecida como Business Model Canvas (BMC) (DORNELAS *et al.*, 2015). Essa ferramenta possibilita que se tenha uma visão geral das quatro áreas básicas para a construção de um negócio: clientes, oferta, infraestrutura e viabilidade financeira (SILVA, 2019). Ela é composta por um quadro que reúne nove blocos que juntos apresentam uma visão geral do negócio (SEBRAE, 2013). A Figura 4.22 apresenta o quadro do modelo canvas. Nela vemos que existem algumas perguntas que devem ser respondidas para se realizar o preenchimento correto desse quadro.

Figura 4.22 | Quadro para a construção do modelo de negócio



Fonte: adaptada de Sebrae (2013).

O valor de um produto é a razão ou o motivo pelo qual as pessoas adquirem os produtos ou serviços oferecidos (SEBRAE, 2013). Desse modo, para o preenchimento do bloco “proposta de valor” é necessário compreender quais as necessidades são atendidas e quais são ganhos obtidos pelos clientes ao utilizaram o produto ou serviço oferecido, ou seja, você deve responder à seguinte pergunta: por que os clientes comprarão de mim e não dos meus concorrentes? A proposta de valor não é o produto que está sendo oferecido, ela representa o valor que esse produto tem para os clientes.

Assimile

Modelo de negócio e plano de negócio são conceitos que podem parecer sinônimos, mas são diferentes. O modelo de negócio corresponde a um passo anterior ao desenvolvimento do plano de negócios. É através do modelo de negócio que é possível identificar se uma ideia inicial tem viabilidade para se transformar em um produto ou serviço. O plano de negócio é mais complexo e demonstra como o negócio será construído com etapas, prazos, planilhas de custos, receitas, funcionários etc. (SEBRAE, 2013).

Na Figura 4.22 também é possível ver os campos Relacionamento com os clientes, segmentos de clientes e canais. Esses campos são os responsáveis por descrever quem é o público-alvo do produto ou serviço oferecido e como a proposta de valor será entregue aos clientes. No bloco “segmentos de clientes” são definidos os clientes a que o produto ou serviço pretende atender. Para o preenchimento desse bloco, podem ser respondidas as perguntas: para quem estamos criando valor? Quem são nossos clientes mais importantes? O bloco “canais” explica a forma como o produto ou serviço deve ser entregue aos clientes. A pergunta que ajuda a compreender a forma pela qual os clientes terão acesso ao produto oferecido é: como o cliente encontrará os produtos e serviços oferecidos? O campo referente ao relacionamento com o cliente demonstra quais as estratégias a serem seguidas para conquistar e manter os clientes.

No bloco correspondente aos “recursos principais” são definidos todos os recursos necessários para se construir a proposta de valor do produto, ou seja, deve ser decidido o que é necessário realizar para que o negócio funcione. Exemplos de perguntas que auxiliam na definição desses recursos são: quais recursos, canais de distribuição e tipos de o relacionamento com clientes serão necessários para viabilizar a proposta de valor? No bloco “atividades principais” são listadas as ações a serem realizadas para que seja possível entregar a proposta de valor ao cliente. Devem ser mostradas as atividades mais importantes para a construção do negócio. Algumas perguntas que ajudam a pensar nas atividades principais são: quais as principais

atividades que nossas propostas de valor exigem? O bloco “parcerias principais” busca identificar os fornecedores e parceiros do negócio. As parcerias buscam otimizar o negócio e reduzir os riscos envolvidos na criação de um empreendimento. As perguntas que auxiliam na identificação de parceiros são: quem são os nossos principais parceiros? Quem são os nossos principais fornecedores?

O bloco “fontes de receita” exibe o modo como o negócio gera receita, ou seja, é necessário definir o quanto e como os clientes pagarão pelo produto ou pelo serviço oferecido. Algumas perguntas que ajudam a compreender as fontes de receita são: que valor os nossos clientes estão realmente dispostos a pagar? Pelo que eles pagam atualmente? Como eles estão pagando atualmente? No bloco “estrutura de custos” são levantados e elencados todos os custos envolvidos na realização do negócio, ou seja, são definidos os gastos necessários para viabilizar a proposta de valor. Exemplos de perguntas que ajudam a definir os custos são: quais são os custos mais representativos em nosso modelo de negócios? Quais são os recursos-chave mais caros? Quais as atividades são as mais caras?

Responder a essas perguntas é essencial para que seja possível ter uma visão geral do negócio. Quando essas perguntas são direcionadas para a análise de viabilidade econômica de aplicações IoT, podem ser identificados alguns pontos-chave em cada um dos blocos do canvas (DIJKMAN *et al.*, 2015). A Figura 4.23 apresenta o canvas com elementos que merecem destaque em cada um dos blocos.

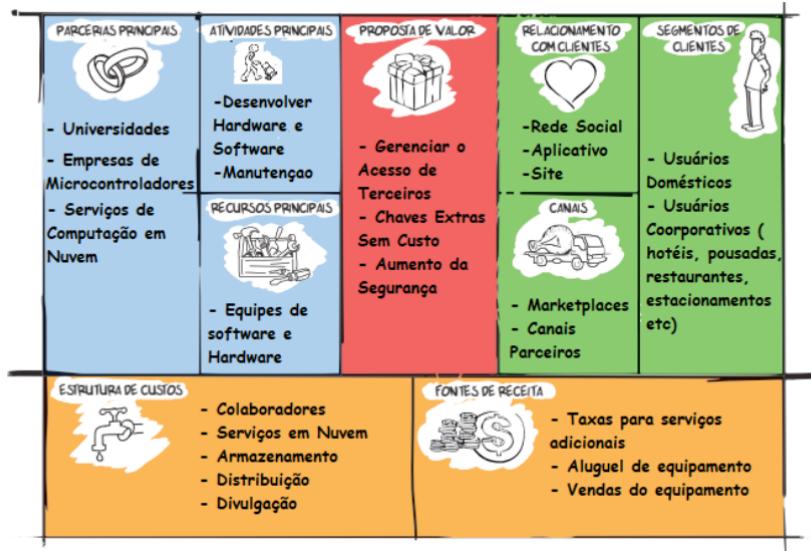
Figura 4.23 | Modelo de negócio para aplicações IoT

Parcerias chave	Atividades chave	Oferta de valor	Relacionamento	Segmentos de clientes
Produtores de Hardware Desenvolvedores de Software Outros Fornecedores Interpretação dos Dados Distribuidores Logística Launching Customers	Desenvolvimento do Produto Desenvolvimento do Software Desenvolvimento dos Consumidores Marketing Vendas Gerenciamento de Parcerias Desenvolvimento de Plataformas	Conveniência/usabilidade Performance Conforto Possibilidade de Upgrades Acessibilidade Redução dos Custos Redução dos Riscos Customização Design Preço Inovação Marca/Status	Comunidades Cocriação Assistência Pessoal Autoserviço (self-service) Assistência Dedicada	Nicho de Mercado Segmentos de Mercado Diversificação Mercado de Massa Plataformas Multilaterais
Recursos chave			Canais	
Recursos Físicos Propriedade Intelectual Recursos Financeiros Software Capacidade dos Funcionários			Vendas Web Lojas de Parceiros Lojas Proprias Atacadistas Força de Vendas	
Estrutura de custos		Fontes de receita		
Custos de Desenvolvimento do Produto Custos com Tecnologia da Informação Custos com Hardware/Produção	Custos com Marketing Custos com Logística	Taxas de Inscrição Taxas de Utilização Venda de Ativos Empréstimo Aluguel	Leasing Licenciamento Propagandas Taxas de Instalação Taxas de Corretagem	

Fonte: adaptada de Dijkman (2015).

A Figura 4.24 apresenta um modelo de preenchimento do quadro canvas para uma fechadura eletrônica inteligente (KLEIN; PACHECO; RIGHI, 2017). Essa fechadura utiliza plataformas abertas e permite que outros usuários utilizem essa fechadura em seus próprios projetos. Os clientes desse produto são hotéis e usuários domésticos que podem, por exemplo, gerar algumas chaves virtuais e permitir o acesso a ambientes durante um período específico de tempo.

Figura 4.24 | Modelo de preenchimento do quadro canvas para uma fechadura eletrônica inteligente



Fonte: adaptada de Klein, Pacheco e Righi (2017).

Agora você já tem o conhecimento técnico e os requisitos necessários para construir aplicações IoT que agreguem valor à sociedade. Não deixe de colocar em prática tudo o que foi aprendido durante esta disciplina. Bons estudos!

Sem medo de errar

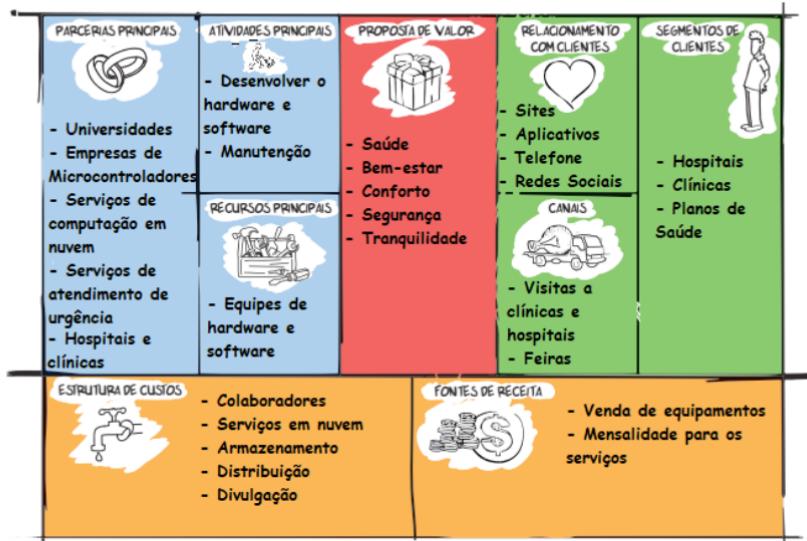
A utilização da IoT em serviços de saúde possibilita um monitoramento remoto, atendimento personalizado, maior acesso aos serviços oferecidos, além de proporcionar maior conforto e independência aos pacientes idosos

e portadores de doenças crônicas. Essas características fazem com que as aplicações IoT na área da saúde estejam entre as mais importantes e com maior valor para a sociedade. Nesse sentido, desenvolver sistemas que propiciem um maior acesso a serviços de saúde de qualidade é fundamental para a construção de um mundo cada vez melhor.

Você está desenvolvendo um sistema de monitoramento remoto de pacientes que possibilita o rápido acesso aos serviços de atendimento de emergência. Depois de indicar os sensores, atuadores e de construir o esquema de toda a aplicação, você precisa terminar o relatório. A etapa final desse relatório consiste em apresentar um modelo de negócios para os diretores da grande rede hospitalar.

O modelo de negócios é fundamental para a avaliação da viabilidade de um projeto. Por meio dele é possível construir o plano de negócios de uma aplicação. Para a construção do modelo de negócios, você pode utilizar o modelo canvas, pelo qual é possível analisar a viabilidade econômica desse projeto e ter uma visão geral do negócio a ser desenvolvido. A Figura 4.25 apresenta uma sugestão de construção do modelo de negócio para o sistema de monitoramento remoto de pacientes.

Figura 4.25 | Exemplo de um possível modelo de negócio para o monitoramento remoto de pacientes

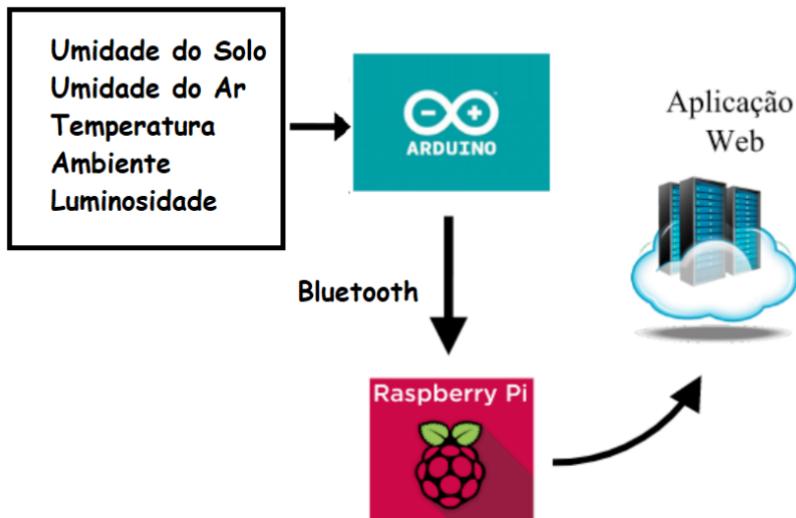


Fonte: elaborada pelo autor.

Monitoramento em casas de vegetação

Você teve a ideia de construir um sistema que possa ser utilizado para o monitoramento do microclima existente em casas de vegetação. Para isso, já definiu os sensores, os atuadores e os demais sistemas necessários para construir essa aplicação, que consiste na utilização de sensores de umidade do solo, de temperatura ambiente, de umidade atmosférica e de luminosidade para monitorar o clima dentro de casas de vegetação. Esses sensores devem estar conectados a um Arduino que deve enviar os dados coletados para um Raspberry Pi (Rpi). A troca de dados entre o Arduino e o Rpi ocorre através do protocolo Bluetooth. A partir do Rpi, os dados precisam ser enviados para servidores em nuvem a fim de possibilitar o acesso remoto das informações. Essas informações podem ser acessadas via web ou por meio de aplicativos de smartphone. A Figura 4.26 apresenta a visão geral da aplicação.

Figura 4.26 | Visão geral da aplicação de monitoramento em casas de vegetação



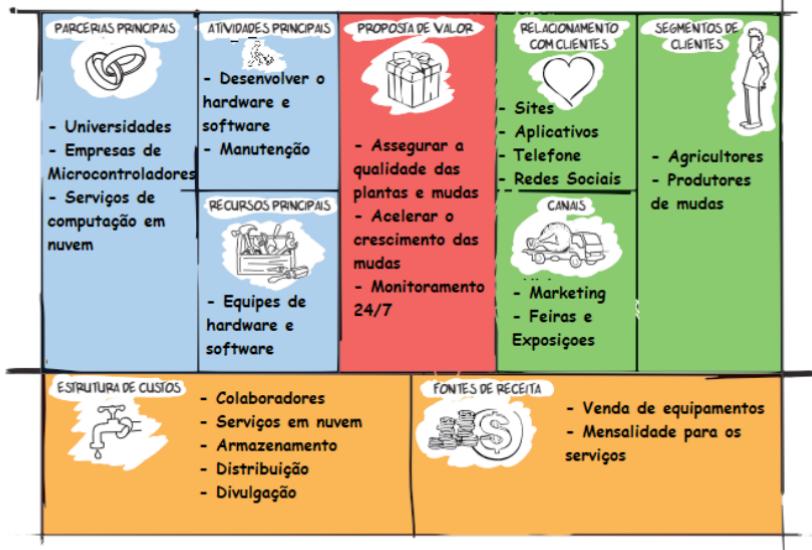
Fonte: adaptada de Ferraz Filho (2018).

Para verificar a viabilidade econômica dessa ideia, é necessário construir um modelo de negócios. A partir desse modelo é possível ter uma visão geral do produto a ser desenvolvido. Assim, você deve construir um modelo de negócios para essa aplicação utilizando o quadro canvas.

Resolução da situação-problema

A Figura 4.27 apresenta uma sugestão de construção do modelo de negócio para o sistema de monitoramento remoto de casas de vegetação.

Figura 4.27 | Exemplo de modelo de negócio para o sistema de monitoramento remoto



Fonte: elaborada pelo autor.

Faça valer a pena

1. A definição da proposta de valor é essencial para indicar o que a empresa oferecerá para o mercado. A proposta de valor não é o produto que será entregue. Ela representa o valor que esse produto trará aos seus clientes.

Com essa definição em mente, considere a seguinte situação. Você e um grupo de amigos tiveram a ideia de construir um semáforo inteligente. Esse semáforo será capaz de contar a quantidade de veículos existentes na via e estimar o fluxo de automóveis em cada período do dia.

Assinale a alternativa que apresenta uma correta indicação da proposta de valor para esse negócio.

- Utilização de redes sociais, website e telefone.
- Empresas de transporte e prefeituras.
- Segurança e agilidade no trânsito. Menos poluição.

- d. Desenvolvedores de software e hardware. Plataformas em nuvem.
- e. Custos de manutenção do equipamento. Equipe de desenvolvimento.

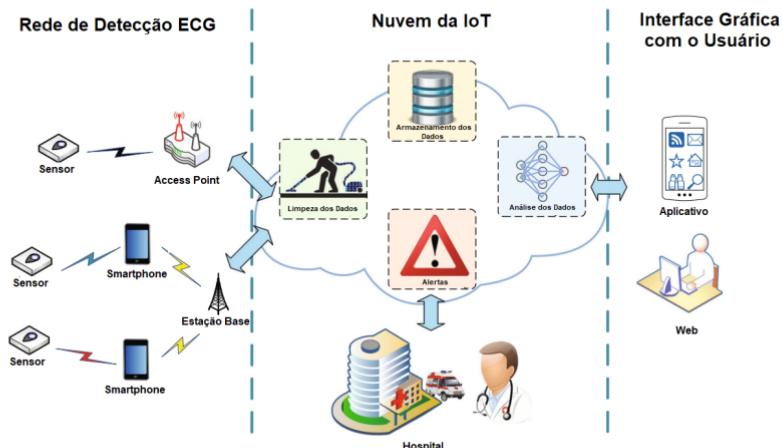
2. Você foi contratado para construir um sistema de monitoramento agrícola para o cultivo de mudas em casas de vegetação. Para isso, foram definidos alguns requisitos. Esse sistema deve ser capaz de obter dados de variáveis como temperatura, umidade do ar e do solo, além de possibilitar o acompanhamento da quantidade de nutrientes no solo. Os dados coletados devem ser disponibilizados através das páginas web e por meio de aplicativos de smartphone. Além disso, devem existir atuadores que possam manter a temperatura ambiente, a umidade do solo e a umidade do ar dentro dos parâmetros estabelecidos pelo administrador do sistema.

Com base no exposto, assinale a alternativa que contém apenas exemplos de atuadores que podem ser utilizados para controlar o microclima dessa casa de vegetação.

- a. Exaustores, aquecedores e umidificadores.
- b. Termômetro, higrômetro e acelerômetro.
- c. DHT22, DHT11 e termômetro.
- d. Barômetro, exaustor e irrigação.
- e. LDR, aquecedores e DHT22.

3. A utilização da IoT para melhoria e universalização dos serviços de saúde corresponde a uma das grandes áreas de aplicação da Internet das Coisas. Um exemplo de utilização da IoT na saúde é o monitoramento remoto do eletrocardiograma (ECG) do paciente (YANG *et al.*, 2016). Para o acompanhamento remoto do ECG são adicionados ao corpo do paciente alguns sensores responsáveis por capturar os dados em tempo real. Esses dados são enviados através da comunicação sem fio para plataformas em nuvem. Nas plataformas em nuvem, esses dados são analisados e, quando anomalias são identificadas, são disparados alertas para os médicos responsáveis. A figura a seguir apresenta a arquitetura descrita.

Figura | Esquema para monitoramento remoto do ECG de pacientes



Fonte: adaptada Yang et al. (2016).

Com base no exposto e na figura apresentada, são realizas as seguintes afirmações sobre esse sistema de monitoramento do ECG de pacientes.

- Para a comunicação sem fio entre os sensores e o smartphone, pode ser utilizado o protocolo Bluetooth.
- Para a troca de mensagens entre o sensor e a nuvem IoT, através do Access Point, pode ser utilizado o Wi-Fi.
- A nuvem IoT não pode ser utilizada para o processamento dos dados, pois o ECG requer alta taxa de aquisição.
- A visualização dos dados através da web é preferível, pois os smartphones não têm alta capacidade de processamento.

Indique qual das alternativas apresenta as afirmações corretas sobre a aplicação apresentada.

- Somente as afirmações I e II estão corretas.
- Somente as afirmações II e III estão corretas.
- Somente as afirmações I e III estão corretas.
- Somente as afirmações I e IV estão corretas.
- Somente as afirmações II e IV estão corretas.

Referências

- ALAM, M.; FERREIRA, J.; FONSECA, J. (Ed.). **Intelligent transportation systems**: dependable vehicular communications for improved road safety. [S.I.]: Springer, 2016.
- AL-NAIMA, F. M.; HAMD, H. A. Vehicle traffic congestion estimation based on rfid. **International Journal of Engineering Business Management**, [s.l.], v. 4, p. 30, 2012.
- ALVES, R. dos S. *et al.* Redes veiculares: princípios, aplicações e desafios. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES, SBRC, 2009, [s.l.]. **Minicursos** [...]. [S.I.]: SBRC, 2009. p. 17-24.
- ARENA, F; PAU, G. An overview of vehicular communications. **Future Internet**, [s.l.], v. 11, n. 2, p. 27, 2019.
- BELTRÃO, N. E. de M.; FILHO, J. F.; FIGUEIRÊDO, I. C. de M. Uso adequado de casa-de-vegetação e de telados na experimentação agrícola. **Revista Brasileira de Engenharia Agrícola e Ambiental**, [s.l.], v. 6, n. 3, p. 547-552, 2002.
- BERNARDO, S. **Impacto ambiental da irrigação no Brasil**. Recursos hídricos e desenvolvimento sustentável da agricultura. Viçosa: MMA, SRH, ABEAS, UFV, 1997. v. 34.
- BOFF, L. Sustentabilidade: tentativa de definição. **Leonardo Boff** [blog]. [S.I.], 15 jan. 2012. Disponível em: <https://leonardoboff.wordpress.com/2012/01/15/sustentabilidade-tentativa-de-definicao/>. Acesso em: 20 set. 2019.
- BONOMI, F. *et al.* Fog computing and its role in the internet of things. In: MCC WORKSHOP ON MOBILE CLOUD COMPUTING, 1., 2012, [s.l.]. **Proceedings** [...]. [S.I.]: ACM, 2012. p. 13-16.
- BORGIA, E. The Internet of Things vision: key features, applications and open issues. **Computer Communications**, [s.l.], v. 54, p. 1-31, 2014.
- BRASIL, A. *et al.* Plano de negócio: uma ferramenta com múltiplas aplicações. **Revista Conexão Eletrônica**, [s.l.], v. 10, n. 1, 2016.
- BRUECK, A. *et al.* A real-time wireless sweat rate measurement system for physical activity monitoring. **Sensors**, [s.l.], v. 18, n. 2, p. 533, 2018.
- BULL, A. **Traffic congestion**: the problem and how to deal with it. [S.I.]: United Nations Publications, 2003.
- CAI, H. *et al.* IoT-based configurable information service platform for product lifecycle management. **IEEE Transactions on Industrial Informatics**, [s.l.], v. 10, n. 2, p. 1558-1567, 2014.
- CAO, Y. *et al.* FAST: A fog computing assisted distributed analytics system to monitor fall for stroke mitigation. In: IEEE INTERNATIONAL CONFERENCE ON NETWORKING, ARCHITECTURE AND STORAGE (NAS), 2015, [s.l.]. **Anais** [...]. [S.I.]: IEEE, 2015. p. 2-11.

COPPOLA, R.; MORISIO, M. Connected car: technologies, issues, future trends. *ACM Computing Surveys (CSUR)*, [s.l.], v. 49, n. 3, p. 46, 2016.

CRUZ, P. et al. SensingBus: using bus lines and fog computing for smart sensing the city. *IEEE Cloud Comput*, [s.l.], p. 1-11, 2018.

DA SILVA, I. J. O. **A rastreabilidade dos produtos agropecuários do Brasil destinados à exportação**. [S.l.: s.n.], 2004.

DAHLMAN, E.; PARKVALL, S.; SKOLD, J. **4G: LTE/LTE-advanced for mobile broadband**. [S.l.]: Academic press, 2013.

DESHPANDE, A.; PITALE, P.; SANAP, S. Industrial automation using Internet of Things (IOT). *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, [s.l.], v. 5, n. 2, p. 266-269, 2016.

DIJKMAN, R. M. et al. Business models for the Internet of Things. *International Journal of Information Management*, [s.l.], v. 35, n. 6, p. 672-678, 2015.

DORNELAS, J. et al. **Plano de negócios com o modelo Canvas**. [S.l.]: Grupo Gen-LTC, 2015.

DORSCH, J. Making buildings smarter. *Semiconductor Engineering*, [S.l.], 2018. Disponível em: <https://semiengineering.com/buildings-get-smarter-through-tech/>. Acesso em: 20 set. 2019.

ERPINNEWS. Fog computing vs edge computing. *Erpinnews*, [S.l.], 2018. Disponível em: <https://erpinnews.com/fog-computing-vs-edge-computing>. Acesso em: 20 set. 2019.

EVERIS. **Everis Connected Car Report**. [S.l.]: Everis, 2015. Disponível em: http://s3-eu-west-1.amazonaws.com/e17r5k-datat1/everis_documents_downloads/everis+connected+car+report.pdf. Acesso em: 18 set. 2019.

FARUOLO, L. B.; MARTHA, P. J. B. **Medição dinâmica da massa de veículos rodoviários em transporte de cargas líquidas para aplicação de trânsito**. [S.l.: s.n.], 2008.

FERRAZ FILHO, B. da S. et al. **Agricultura de precisão em casas de vegetação**: controle e gestão de cultivo em produção de mudas. [S.l.: s.n.], 2018.

FORD. **Smart cars meet smart homes**: Ford exploring SYNC integration with Amazon Echo and Alexa Wink. Las Vegas, 2016. Disponível em: <https://media.ford.com/content/fordmedia/fna/us/en/news/2016/01/05/smart-cars-meet-smart-homes.html>. Acesso em: 18 set. 2019.

GAUR, A. et al. Smart city architecture and its applications based on IoT. *Procedia Computer Science*, [s.l.], v. 52, p. 1.089-1.094, 2015.

GELLINGS, C. W. **The smart grid**: enabling energy efficiency and demand response. [S.l.]: The Fairmont Press, Inc., 2009.

GHIMIRE, D. et al. Real-time sleepiness detection for driver state monitoring system. *Adv. Sci. Technol. Lett.*, [s.l.], v. 120, p. 1-8, 2015.

GONDCHAWAR, N.; KAWITKAR, R. S. IoT based smart agriculture. **International Journal of advanced research in Computer and Communication Engineering**, [s.l.], v. 5, n. 6, p. 838-842, 2016.

GREENGARD, S. Automotive systems get smarter. **Communications of the ACM**, [s.l.], v. 58, n. 10, p. 18-20, 2015.

GREENIQ Smart Garden Hub - Control Your Garden with Your Smartphone. [S. l.: s. n.], 2018. 1 vídeo (2 min 12 s). Publicado pelo canal GreenIQ. Disponível em: <https://www.youtube.com/watch?v=LEdmtUrzDKs>. Acesso em: 23 set. 2019.

GREER, C. *et al.* **Nist framework and roadmap for smart grid interoperability standards, release 3.0**. [S.l.: s.n.], 2014.

GUERRERO-IBÁÑEZ, J.; ZEADALLY, S.; CONTRERAS-CASTILLO, J. Sensor technologies for intelligent transportation systems. **Sensors**, [s.l.], v. 18, n. 4, p. 1.212, 2018.

HANK, P. *et al.* Automotive ethernet: in-vehicle networking and smart mobility. In: CONFERENCE ON DESIGN, AUTOMATION AND TEST IN EUROPE, 2013, [s.l.]. **Proceedings** [...]. [S.l.]: EDA Consortium, 2013. p. 1.735-1.739.

HAYAMA, K.; MINAKATA, T. A Far Infrared Vehicle Sensor for a Traffic Signal Control. In: WORLD CONGRESS ON INTELLIGENT TRANSPORT SYSTEMS AND ITS AMERICA'S 2008 ANNUAL MEETING ITS AMERICA ERTICOITS JAPAN TRANSCORE, 15., 2008, [s.l.]. **Anais** [...]. [S.l.: s.n.], 2008.

HU, F.; XIE, D.; SHEN, S. On the application of the internet of things in the field of medical and health care. In: INTERNATIONAL CONFERENCE ON GREEN COMPUTING AND COMMUNICATIONS AND IEEE INTERNET OF THINGS AND IEEE CYBER, PHYSICAL AND SOCIAL COMPUTING, 2013, [s.l.]. **Proceedings** [...]. [S.l.]: IEEE, 2013. p. 2.053-2.058.

HUA, B. D. C. *et al.* Internet of Things (IOT) Monitoring System for Elderly. In: INTERNATIONAL CONFERENCE ON INTELLIGENT AND ADVANCED SYSTEM (ICIAS), 2018, [s.l.]. **Proceedings** [...]. [S.l.]: IEEE, 2018. p. 1-6.

ISLAM, S. M. R. *et al.* The internet of things for health care: a comprehensive survey. **IEEE Access**, [s.l.], v. 3, p. 678-708, 2015.

KAMILARIS, A. *et al.* Agri-IoT: a semantic framework for Internet of Things – enabled smart farming applications. In: WORLD FORUM ON INTERNET OF THINGS (WF-IoT), 3., 2016, [s.l.]. **Proceedings** [...]. [S.l.]: IEEE, 2016. p. 442-447.

KHAIRNAR, M.; VAISHALI, D.; PRADHAN, Dr SN. **V2V communication survey wireless technology**. arXiv preprint arXiv: 1403.3993, 2014.

KLEIN, A.; PACHECO, F. B.; RIGHI, R. da R. Internet of things-based products/services: process and challenges on developing the business models. **JISTEM-Journal of Information Systems and Technology Management**, [s.l.], v. 14, n. 3, p. 439-461, 2017.

KONDEPUDI, S. N. *et al.* **Smart sustainable cities analysis of definitions.** The ITU-T focus group for smart sustainable cities. [S.l.: s.n.], 2014.

KORKAS, C. D. *et al.* Intelligent energy and thermal comfort management in grid-connected microgrids with heterogeneous occupancy schedule. **Applied Energy**, [s.l.], v. 149, p. 194-203, 2015.

LACINÁK, M.; RISTVEJ, J. Smart city, safety and security. **Procedia Engineering**, [s.l.], v. 192, p. 522-527, 2017.

LETSHWITI, V.; LAMPRECHT, T. J. **Appropriate technology for automatic passenger counting on public transport vehicles in South Africa.** [S.l.]: SATC, 2004.

LIU, L. IoT and A Sustainable City. **Energy Procedia**, v. 153, p. 342-346, 2018.

LU, N. *et al.* Connected vehicles: solutions and challenges. **IEEE internet of things journal**, [s.l.], v. 1, n. 4, p. 289-299, 2014.

LUAN, T. H. *et al.* Social on the road: enabling secure and efficient social networking on highways. **IEEE Wireless Communications**, [s.l.], v. 22, n. 1, p. 44-51, 2015.

MANYIKA, J. *et al.* **Unlocking the potential of the Internet of Things.** [S.l.]: McKinsey Global Institute, 2015.

MICROSOFT. Saving lives by alerting people about earthquakes before they strike. **Microsoft: customer stories.** [S.l.], 2017. Disponível em: <https://customers.microsoft.com/en-us/story/sky-alert>. Acesso em: 20 set. 2019.

MINOLI, D.; SOHRABY, K.; OCCHIOGROSSO, B. IoT considerations, requirements, and architectures for smart buildings – energy optimization and next-generation building management systems. **IEEE Internet of Things Journal**, [s.l.], v. 4, n. 1, p. 269-283, 2017.

MIUCIC, R. (Ed.). **Connected vehicles:** Intelligent Transportation Systems. [S.l.]: Springer, 2018.

MOHANTY, S. P.; CHOPPALI, U.; KOUGIANOS, E. Everything you wanted to know about smart cities: The Internet of Things is the backbone. **IEEE Consumer Electronics Magazine**, [s.l.], v. 5, n. 3, p. 60-70, 2016.

MORVAJ, B.; LUGARIC, L.; KRAJCAR, S. Demonstrating smart buildings and smart grid features in a smart energy city. In: INTERNATIONAL YOUTH CONFERENCE ON ENERGETICS (IYCE), 3., 2011, [s.l.]. **Proceedings** [...]. [S.l.]: IEEE, 2011. p. 1-8.

MUANGPRATHUB, J. *et al.* IoT and agriculture data analysis for smart farm. **Computers and electronics in agriculture**, [s.l.], v. 156, p. 467-474, 2019.

MUTHURAMALINGAM, S. *et al.* IoT Based Intelligent Transportation System (IoT-ITS) for Global Perspective: A Case Study. In: INTERNET OF THINGS AND BIG DATA ANALYTICS FOR SMART GENERATION, 2009, Cham. **Anais** [...]. Cham: Springer, 2019. p. 279-300.

PALA, Z.; INANC, N. Smart parking applications using RFID technology. In: ANNUAL RFID EURASIA, 1., 2007. *Anais* [...]. [s.l.]: IEEE, 2007. p. 1-3.

PANDIT, A. A.; MUNDRA, A. K.; TALREJA, J. RFID tracking system for vehicles (RTSV). In: INTERNATIONAL CONFERENCE ON COMPUTATIONAL INTELLIGENCE, COMMUNICATION SYSTEMS AND NETWORKS, 1., 2009, [s.l.]. *Anais* [...]. [s.l.]: IEEE, 2009. p. 160-165.

PANG, Z. **Technologies and Architectures of the Internet-of-Things (IoT) for Health and Well-being**. 2013. Tese (Doutorado em Sistemas Eletrônicos e Computacionais) – KTH Royal Institute of Technology, Estocolmo, Suécia, 2013.

PARRADO, N.; DONOSO, Y. Congestion based mechanism for route discovery in a V2I-V2V system applying smart devices and IoT. *Sensors*, [s.l.], v. 15, n. 4, p. 7.768-7.806, 2015.

PERERA, C. et al. Fog computing for sustainable smart cities: A survey. *ACM Computing Surveys (CSUR)*, [s.l.], v. 50, n. 3, p. 32, 2017.

QUANTOS metros de fiação existe em um carro? **Quatro Rodas**. [s.l.], 2017. Disponível em: <https://quatrorodas.abril.com.br/auto-servico/quantos-metros-de-fiacao-existe-em-um-carro/>. Acesso em: 18 set. 2019.

RAJI, A.; JEYASHEELI, P. G.; JENITHA, T. IoT based classification of vital signs data for chronic disease monitoring. In: INTERNATIONAL CONFERENCE ON INTELLIGENT SYSTEMS AND CONTROL (ISCO), 10., 2016, [s.l.]. *Proceedings* [...]. [s.l.]: IEEE, 2016. p. 1-5.

RANDALL, T. The smartest building in the world. **Bloomberg**. [s.l.], 2015. Disponível em: <https://www.bloomberg.com/features/2015-the-edge-the-worlds-greenest-building/>. Acesso em: 20 set. 2019.

RANI, S. U. et al. Iot Patient Health Monitoring System. *Indian Journal of Public Health Research & Development*, [s.l.], v. 8, n. 4, 2017.

ROZA, D. Novidade no campo: geotecnologias renovam a agricultura. *Revista InfoGEO*, [s.l.], n. 11, jan./fev., 2000.

SCHULZ, P. et al. Latency critical iot applications in 5G: perspective on the design of radio interface and network architecture. *IEEE Communications Magazine*, [s.l.], v. 55, n. 2, p. 70-78, 2017.

SCHUMACHER, J.; GSCHWEIDL, M.; RIEDER, M. EURIDICE – an enabler for intelligent cargo for the logistics sector. *Journal of Systemics, Cybernetics and Informatics*, [s.l.], p. 18-28, 2010.

SERVIÇO BRASILEIRO DE APOIO ÀS MICRO E PEQUENAS EMPRESAS (Sebrae). **O quadro de modelo de negócios** – cartilha. Brasília, DF: Sebrae, 2013.

SHARMA, A. K. et al. IOT enabled forest fire detection and online monitoring system. *Int J Curr Trends Eng Res (IJCTER)*, [s.l.], v. 3, n. 5, p. 50-54, 2017.

SHI, W. *et al.* Edge computing: Vision and challenges. **IEEE Internet of Things Journal**, [s.l.], v. 3, n. 5, p. 637-646, 2016.

SILVA, A. F. da. O que é business model canvas e como fazer um? **Guia Empreendedor**. [S.l.], 2019. Disponível em: <https://guiaempreendedor.com/o-que-business-model-canvas/>. Acesso em: 20 set. 2019.

SIVAKUMAR, R.; MANGALAM, H. RADAR based vehicle collision avoidance system used in four wheeler automobile segments. **International Journal of Scientific & Engineering Research**, [s.l.], v. 5, n. 1, 2014.

SIVARAMAN, V. *et al.* Network-level security and privacy control for smart-home IoT devices. In: INTERNATIONAL CONFERENCE ON WIRELESS AND MOBILE COMPUTING, NETWORKING AND COMMUNICATIONS (WiMob), 11., 2015, [s.l]. **Proceedings** [...]. [S.l.]: IEEE, 2015. p. 163-167.

SMOOT, S. R.; TAN, N. K. **Private cloud computing**: consolidation, virtualization, and service-oriented infrastructure. [S.l.]: Elsevier, 2011.

SOMASHEKHAR, G. C.; SHIRABADAGI, S.; HEGADI, R. S. High Density traffic management using image background subtraction algorithm. **International Journal of Computer Applications**, Bangalore, p. 10-15, fev. 2014.

SONI, N. B.; SARASWAT, J. A review of IoT devices for traffic management system. In: INTERNATIONAL CONFERENCE ON INTELLIGENT SUSTAINABLE SYSTEMS (ICISS), 2017, [s.l]. **Anais** [...]. [S.l.]: IEEE, 2017. p. 1.052-1.055.

STEVANOVIC, A. **Adaptive traffic control systems**: domestic and foreign state of practice. [S.l.: s.n.], 2010.

TESCH, M. IoT and predictive analytics: fog and edge computing for industries versus cloud. 2018. **leanBI**. [S.l.], 2018. Disponível em: <https://leanbi.ch/en/blog/iot-and-predictive-analytics-fog-and-edge-computing-for-industries-versus-cloud-19-1-2018/>. Acesso em: 20 set. 2019.

TSAI, C.-W. *et al.* Data mining for internet of things: a survey. **IEEE Communications Surveys and Tutorials**, [s.l.], v. 16, n. 1, p. 77-97, 2014.

TSCHIEDEL, M.; FERREIRA, M. F. Introdução à agricultura de precisão: conceitos e vantagens. **Ciência Rural**, [s.l.], v. 32, n. 1, p. 159-163, 2002.

UHLEMANN, E. Introducing connected vehicles [connected vehicles]. **IEEE Vehicular Technology Magazine**, [s.l.], v. 10, n. 1, p. 23-31, 2015.

UN, D. **World urbanization prospects**: the 2018 revision. Nova York: United Nations Department of Economics and Social Affairs, Population Division, 2018.

VAHIDI, A.; ESKANDARIAN, A. Research advances in intelligent collision avoidance and adaptive cruise control. **IEEE Transactions on Intelligent Transportation Systems**, v. 4, n. 3, p. 143-153, 2003.

VIEIRA, T. P.; ALMEIDA, P. E. M.; MEIRELES, M. R. G. Intelligent fault management system for wireless sensor networks with reduction of power consumption. In: INTERNATIONAL SYMPOSIUM ON INDUSTRIAL ELECTRONICS (ISIE), 26., 2017, [s.l]. **Proceedings** [...]. [S.l.]: IEEE, 2017. p. 1.521-1.527.

WEISER, M. The Computer for the 21 st Century. **Scientific American**, [s.l.], v. 265, n. 3, p. 94-105, 1991.

WENG, T.; AGARWAL, Y. From buildings to smart buildings – sensing and actuation to improve energy efficiency. **IEEE Design & Test of Computers**, [s.l.], v. 29, n. 4, p. 36-44, 2012.

WORLD'S Greenest Office Building Is Dutch: The Edge. Amsterdan: Bloomberg Business, 2015. 1 video (4 min 36 s). Publicado pelo canal Bloomberg. Disponível em: <https://www.youtube.com/watch?v=JSzko-K7dzo>. Acesso em: 19 set. 2019.

WU, M.; SHIH, M. Simulated and experimental study of hydraulic anti-lock braking system using sliding-mode PWM control. **Mechatronics**, [s.l.], v. 13, n. 4, p. 331-351, 2003.

YANG, Z. *et al.* An IoT-cloud based wearable ECG monitoring system for smart healthcare. **Journal of Medical Systems**, [s.l.], v. 40, n. 12, p. 286, 2016.

YANNUZZI, M. *et al.* Key ingredients in an IoT recipe: Fog Computing, Cloud computing, and more Fog Computing. In: INTERNATIONAL WORKSHOP ON COMPUTER AIDED MODELING AND DESIGN OF COMMUNICATION LINKS AND NETWORKS (CAMAD), 19., 2014, [s.l]. **Proceedings** [...]. [S.l.]: IEEE, 2014. p. 325-329.

ISBN 978-85-522-1687-2



9 788552 216872 >