

Internet das Coisas (IoT)



Introdução

Bem-vindo(a) ao emocionante universo da Internet das Coisas (IoT) e sua poderosa aliada, a linguagem programação *Python*. Imagine um mundo onde objetos do dia-a-dia se tornam inteligentes, capazes de se comunicar e tomar decisões por “conta própria” melhor dizendo, decisões inteligentes - Isso é a IoT!

A Internet das Coisas ao ser integrada com *Python*, a linguagem que está impulsionando uma revolução graças a sua facilidade de compreensão e imensa disponibilidade de aplicação, permitem explorar o melhor desempenho de ambas as tecnologias para a construção de ecossistemas complexos, capacitando você a interagir e criar dispositivos inteligentes e soluções inovadoras que podem transformar a maneira como vivemos, trabalhamos e interagimos com o mundo ao nosso redor.

Ficou curioso(a)? Então vamos juntos mergulhar no futuro da conectividade!

INTERNET DAS COISAS (IoT)

Entende-se por Internet das Coisas ou IoT - *Internet of Things*, “um conjunto de tecnologias integradas” sendo que este pode ser utilizado em inúmeras aplicações de diversas áreas do conhecimento: desde a área de transportes (como método de gestão de frota exemplo), agricultura e outras áreas da indústria (com métodos de gestão de custos eficientes, por exemplo), ou até mesmo em real mais comuns como sua aplicação em residências (com sistemas de automatização, mais conhecidos como sistemas para casas inteligentes), saúde e outras atividades e contextos do cotidiano.



Figura 1. Ilustra como o sistema IoT pode contribuir para a agricultura, desde a medição de percentuais, até o cálculo de probabilidades.



Figura 2. Outro exemplo de sistema IoT presente em nosso cotidiano: alto-falante inteligente para controle de atividades domésticas

O termo “Internet das Coisas” foi introduzido pelo professor Kevin Ashton, pesquisador do [MIT - Massachusetts Institute of Technology](#)¹ em uma apresentação realizada em 1999, cujo objetivo era o de nomear “um sistema de sensores capazes de conectar o mundo físico com a Internet”. Para ele, era importante batizar o projeto sem a utilização da palavra “inteligente”, pois a considerava inadequada, e percebeu que precisava ser um nome simples para que os futuros investidores - que pouco entendiam sobre a rede mundial de computadores - compreendessem exatamente o que aquela inovação significaria. Ashton, enfim, percebeu que aquele era um sistema que conectava várias “coisas” e, devido a esse princípio elementar, o definiu então como *Internet of Things* (com “of” em minúsculo, tendo recebido a tradução literal para o idioma português como Internet das Coisas). A apresentação de Kevin foi um sucesso e, desde então, esse sistema de interconexão vem sendo pesquisado e atualizado nos laboratórios de pesquisas do MIT em especial no Auto-ID Lab <<https://autoid.mit.edu/>>, o qual criou as bases dessa tecnologia posteriormente difundida de forma aprimorada ao mundo.

Vale ressaltar também que a ideia de criar uma rede capaz de conectar vários aparelhos diferentes em um único sistema pode ser considerada antiga, mas necessitou de vários anos de pesquisas para que o termo se popularizasse e o assunto realmente começasse a ser discutido e aprimorado pela ciência. O motivo é muito simples: a ideia era muito boa, mas a tecnologia desenvolvida até então era insuficiente para possibilitar o desenvolvimento e aplicação do IoT, afinal, para a Internet das Coisas operar é necessário ter sistemas capazes de conectarem dispositivos com interfaces que garantam a automatização.



Figura 3. Atualmente a IoT está presente quase em sua totalidade em nosso cotidiano: desde a possibilidade de conexão entre eletrônicos utilizados na comunicação a de execução de tarefas simples do dia-a-dia, como lavar roupas utilizando o aplicativo da máquina de lavar através de um dispositivo móvel

Com base em experiências realizadas por meio do sistema de identificação por rádio frequência RF - *Radio-Frequency*, mais especificamente com o RFID - *Radio-Frequency Identification* (tecnologia desenvolvida no início da década de 90, cuja intenção principal é a de monitorar e controlar dados coletados em um sistema controlado) e o desenvolvimento exponencial da Internet, o conceito democrático e generalizado de obter um melhor controle por meio da automação aprimorou o conceito de IoT.

Tal processo de aprimoramento viria anos depois possibilitar a interconexão entre diversos tipos de equipamentos e aparelhos: eletrodomésticos, veículos motores, dispositivos eletrônicos variados, bem como partes de infraestruturas domiciliares como fechaduras, lâmpadas e até mesmo itens vestíveis (*Wearable Technology* ou Tecnologia Vestível) de uso pessoal. Com praticamente tudo o que

puder imaginar, emergindo do avanço de várias áreas correlatas, unificadas aos sistemas embarcados com microeletrônica, programadas, comunicações e principalmente, o uso de sensores e infravermelhos.



Figura 4. Exemplo da IoT em itens de vestuário, os quais permitem a interconexão entre diferentes dispositivos mobile

¹ Conheça vários projetos inovadores mantidos e em pesquisa atualmente pelo MIT na área da IoT
<<https://www.media.mit.edu/research/?filter=everything&tag=internet-things>>

Aplicações

A Internet móvel 5G (Internet de 5^a Geração) em conjunto com a automação de residências e setores da indústria gera um contato frequente com esse tipo de tecnologia em nosso cotidiano, isso ocorre, pois, a IoT permite que diferentes tipos de dispositivos estabeleçam e mantenham uma conexão com a Internet, considerando desde sensores, até eletrodomésticos e automóveis. Pode estar atualmente ser utilizada em dispositivos “coisas” que até pouco tempo atrás não apresentavam essa capacidade de interconexão, simbolizando o estabelecimento de um ambiente virtual integrado, como uma rede ou ecossistema de dispositivos que recebem e transferem dados a todo momento, utilizando *Cloud Computing* (recursos em nuvem) e dessa forma, podem trabalhar em conjunto e maneira integrada.

Para exemplificar a aplicação da IoT na realidade de sistemas aos quais as pessoas interagem na sociedade contemporânea podemos citar, residências automatizadas em que há um tipo de termostato que ajusta a temperatura de acordo com a preferência do morador. O sistema também pode controlar a operação de uma cafeteira sincronizada com o horário do despertador de manhã, para preparar o café da manhã. Também pode haver ainda uma câmera de monitoramento com acesso por celular, permitindo o monitoramento das imagens das dependências residenciais em tempo real e ainda a abertura e/ou fechamento de fechaduras utilizando Apps.

Até aqui ficou claro que a utilização da IoT na realidade cotidiana de residências automatizadas, pode abranger desde o uso de ferramentas comuns, como a troca de dados entre aparelhos (sensores de proximidade e temperatura), até o funcionamento automático de eletrodomésticos pré-programados para funcionarem à distância. Mas para que isso seja possível, obviamente as “coisas” precisam estar conectadas com a Internet a partir de um sinal com dados móveis disponíveis por cabo ou Wi-Fi - Wireless Fidelity, sensores infravermelhos ou sinal Bluetooth (como aplicativos para celular que simulam controles remotos).

Vale ressaltar que poderá levar um longo período em anos até que todos os aparelhos de uso domiciliar desejados pelos consumidores ganhem versões inteligentes e, principalmente, que esses modelos sejam de fato acessíveis ao mercado consumidor geral.



Figura 5. Imagem que ilustra a conexão entre dispositivo eletrônico móvel (celular por App) e eletrodoméstico (cafeteira) através da IoT

Para além da aplicação da IoT em residências, outro campo importante se dá no ambiente industrial. Onde os sistemas de autor contribuem fortemente nos polos industriais para ampliar setores de fabricação e logística, para coordenar cadeias de suprimentos, controlar estoques e linhas de produção.

Nesses casos, por exemplo, podem haver sensores instalados ao longo da linha de produção, os quais detectam produtos defeituados e imediatamente os separam dos demais (contribuindo para elevar os padrões de qualidade da marca em questão). Além disso, também existem projetos recentes de implementação da IoT em escalas maiores, como nas “*Smart Cities* ¹”, cujo desenvolvimento tecnológico capaz de conectar coisas e pessoas em uma escala macro (do usuário à dimensão urbana), e que se de fato implementada de forma adequada, pode promover rápido progresso social e econômico.



Figura 6. Imagem ilustrativa do ecossistema de uma *Smart City*

¹ São cidades que têm na tecnologia um de seus carro-chefe, essas cidades buscam proporcionar um ambiente urbano que promova entre outras coisas, soluções sustentabilidade, para a segurança pública e para o desenvolvimento humano, utilizando seus recursos naturais de forma sustentável e impulsionando a sua economia por meio do desenvolvimento tecnológico consciente.

Desafios

Embora a IoT seja uma inovação positiva, a implementação de tal tecnologia possui alto grau de complexidade e demanda grandes investimentos para a disponibilização de estruturas mais robustas e interconectadas, abrangendo desde uma simples lâmpada inteligente capaz de ser controlada por comandos de voz ou via aplicativos, por exemplo, até a implementação de um ecossistema complexo como uma cidade inteligente, para a qual podem ser requeridos milhares de equipamentos a um bastante custo elevado.

Além disso, há também questões relacionadas à segurança das informações contidas nesse sistema, onde dispositivos de teste gratuitos podem ser considerados os mais fracos e/ou desprotegidos, os quais são altamente sujeitos a invasões através da rede. Nesses casos, há maiores chances desses dispositivos estarem vulneráveis a ataques, comprometendo a atividade e a integridade de outros aparelhos conectados à mesma rede. Em resumo, a IoT já é uma realidade presente, mas a sua aplicação segura depende de muitos fatores relacionados à estrutura, segurança e também à parte econômica, que devem ser sempre priorizados para sua aplicação segura e eficaz.

Outro ponto importante quando se trata da IoT, são os dispositivos que possibilitarão a interconexão das inúmeras “coisas” presentes no ecossistema, conhecidos como sensores de microprocessamento, dispositivos para armazenamento, identificação, controles, softwares para interfaces, além dos sistemas operacionais que garantem a boa usabilidade e a conectividade da Internet, entre ainda outros sistemas de rede aos equipamentos.

Importante frisar que “estar conectado” não significa necessariamente “ser inteligente” do ponto de vista da IoT, pois para que um dispositivo seja realmente “inteligente”, ele deve ser capaz de obter dados, analisar automaticamente esses dados e executar ações consideradas inteligentes sem a intervenção de um ser humano para que tal atividade seja executada. Baseado nessa análise, um dispositivo considerado “inteligente” não necessita que um humano esteja envolvido na sua execução, pois esse processo é automático.



Figura 7. Exemplificação de alguns usos da IoT numa Smart City

Tipos de Sensores e Dispositivos

Os sensores são dispositivos que reagem a estímulos externos, gerando ações pré-programadas para suas funções e cada sensor tem sua função pré-determinada. Os sensores são de extrema importância para conectar as “coisas”.

São alguns exemplos: sensores de proximidade (como os utilizados em portas automatizadas, que abrem quando as pessoas se aproximam), sensores de vibração, sensores de rádio (identificação por RF ou [NFC - Near Field Communication](#)¹), sensores magnéticos, sensores elétricos, sensores químicos de gás (muito utilizados para identificar altos níveis de gás na atmosfera, e que emitem sinais sonoros como aviso quando a quantidade detectada estiver fora de parâmetros especificados), sensores de temperatura, luz e também de velocidade. No contexto da eletrônica, um sensor comumente utilizado é o de monóxido de carbono MQ-7, o qual é aplicado em projetos do tipo *Arduino* <<https://www.arduino.cc/>>.



Figura 8. Exemplo físico de um sensor MQ-7

Os tipos de sensores mais utilizados no mercado para implementação da IoT são os de **proximidade** devido à sua simples utilização e alta eficácia, como, por exemplo, nas seguintes situações:

- Situação 1 - Evitar a proximidade de pessoas ou animais a áreas restritas; e

- Situação 2 - Indicar visualmente quais vagas estão disponíveis e quais estão ocupadas em estacionamentos de *shoppings*, aeroportos e supermercados, podendo integrar um sistema de contagem e controle do fluxo de carros nos estacionamentos através de inteligência e conectividade.

Há também os **acelerômetros**, sensores cuja função é medir a variação de aceleração e vibração de determinado equipamento ou estruturas, como os para: celulares, joysticks de videogames, betoneiras na construção civil, geradores em sistemas de infraestrutura elétrica, dentre outros.

Já os sensores de **temperatura** e **umidade** são os mais populares em relação à Internet das Coisas. Através deles é possível identificar a **umidade** relativa do ar por exemplo, além de aferir e controlar **temperaturas** corporais (com sua utilização voltada para área da saúde), de um forno ou até mesmo de uma caldeira industrial (com sua utilização voltada para operações e produção em pequenas e grandes escala), controle de perecíveis em transportes ou locais de armazenagem, além de outras aplicações.

Outro exemplo de sensor, o qual está entre os mais conhecidos e utilizados é o **GPS - Global Positioning System** ou Sistema de Posicionamento Global. Esse sensor vem instalado de fábrica em praticamente todos os tipos de dispositivos móveis e sistemas inteligentes de automóveis fabricados nos dias atuais, auxiliando a identificar nossa posição geográfica e permitindo a utilização de aplicativos de trânsito e navegação. Esse tipo de sensor permite ao dispositivo localizar qualquer objeto ou indivíduo que também pode ser monitorado por um sensor **GPS** via satélite.



Figura 9. Painel digital interativo em automóvel, o qual possibilita a utilização do **GPS** em tempo real.

Em conjunto com os **sensores**, podemos incluir também os **atuadores**, que são responsáveis por auxiliar os movimentos mecânicos que melhoram a ação dos sensores em relação ao ambiente. Portanto, saber trabalhar com os sensores e conhecer as suas diversas categorias faz-se essencial para o trabalho com equipamentos que se conectam aos sistemas da IoT.

Saiba Mais

Para conhecer mais sobre sensores, seus diferentes tipos e utilidades, assista ao vídeo “Sensores usados na Internet das Coisas (IoT)” disponível no YouTube através do link :

<<https://www.youtube.com/watch?v=E-7NvazztZo&t=304s>>

¹Near-Field Communication ou Comunicação por Campo de Proximidade, é uma tecnologia que permite a troca de informações sem fio, de forma segura entre dois dispositivos compatíveis que estejam próximos sem a necessidade de ter configurações prévias. Entre as funções NFC mais comuns estão: o pagamento de contas, compra ou recarga de cartão de passagens de ônibus, trens, metrôs, balsas, etc. Sendo possível ainda compartilhar arquivos e parear dispositivos acessórios.

Introdução Prática

E qual é a linguagem de programação utilizada em IoT? Bom, é necessário ter linguagens “leves”, para não comprometer os espaço de armazenamento dos dispositivos, assim não sendo necessário equipamentos tão robustos com maiores custos, além de algumas limitações de *hardware*. Ou seja, é importante desenvolver linguagens que não consumam muita memória e processamento do equipamento, o que nos leva à linguagem C++.

C++

É a principal linguagem utilizada em IoT por ser leve, permitindo aos desenvolvedores melhores possibilidades de mudanças e de das funções de certos dispositivos IoT, ela possui rápida execução das funções pré-determinadas, o que gera baixos consumos de energia, economia de memória e processamento dos equipamentos. Suas desvantagens são: as grandes linhas de códigos para execução de tarefas em geral, somada a linguagem ser de difícil compreensão, podendo gerar erros consecutivos. Nos casos de desenvolvimento de infraestruturas mais complexas e robustas, visando investir em equipamentos de alta performance, é possível o pela linguagem de programação Python.

Python

Considerada a linguagem de programação mais simples para o desenvolvimento de *scripts* e de mais fácil compreensão, Python tornou versátil e abrangente para execução de várias funções, sejam elas cotidianas ou específicas.

A simplicidade e praticidade de uso foram capazes de tornar o acesso de suas bibliotecas nos três principais SOs: *MacOS*, *MS-Windows* e *Linux* (com algumas alterações nos códigos de acesso as bibliotecas podem variar de sistema para sistema).

Vale ressaltar que, mesmo sendo uma das linguagens de programação mais utilizadas no mundo, suas funções para IoT podem ligeiramente limitadas, pois, como sua linguagem é de alto nível, alguns equipamentos podem não funcionar bem por serem limitados suas configurações de *hardware*, o que pode acarretar na ocorrência de "bugs" no sistema, como travamentos e má execução das funções do *script*.

Tendo em vista esses aspectos, é importante enfatizar que antes de escolher a linguagem utilizada no sistema, é importante buscar orientação quanto ao porte do equipamento selecionado e o quanto poderá ser investido financeiramente no projeto.

De qualquer forma, Python figura como a melhor opção entre as duas para programação em IoT e automação, pois sua filosofia estrutural, comunidade fortemente ativa e a grande quantidade de bibliotecas especializadas, tornam muito mais simples a implementação de aplicações em plataformas como *Arduino* e *Raspberry Pi*.

Para Aprender e Treinar

Um ótimo exemplo para fixação do conteúdo exposto até aqui, é utilizá-lo em aplicações do dia-a-dia. Provavelmente a aplicação comum utilizando a IoT, seja a de acender ou apagar uma lâmpada, ou no caso, um LED - *Light Emitter Diode* ou Diodo Emissor de Luz.

Para o experimento que faremos, propomos a utilização de uma plataforma bastante conhecida chamada *Tinkercad* da empresa Autodesk <<https://www.tinkercad.com/>>, essa plataforma disponibiliza um ambiente virtual para simulações de circuitos, protótipos 3D e códigos em bloco com os quais é possível projetar, testar e implementar inúmeras aplicações.



Figura 10. Fragmento inicial da página oficial do Autodesk *Tinkercad*

A plataforma disponibiliza uma bancada virtual, na qual podem ser simulados diferentes circuitos eletrônicos, ou seja, para IoT e outras áreas. A bancada funciona como um simulador de experimentos reais, permitindo que o desenvolvedor busque em sua base, componentes diversos de forma categorizada e que em situações como quando da utilização do Arduino, permite a interação via linguagem de programação.

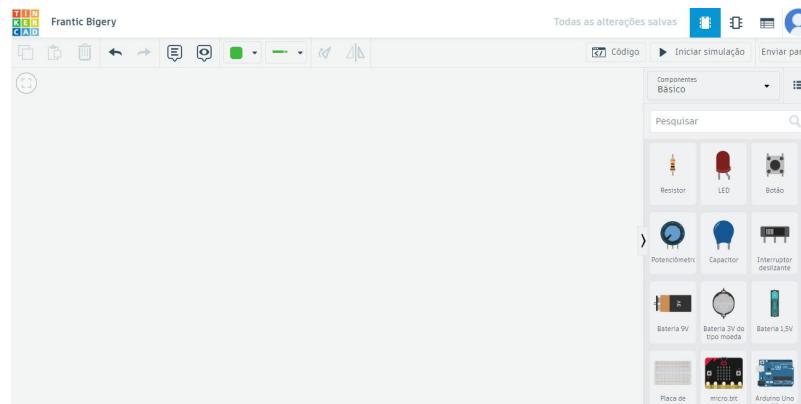


Figura 11. Demonstração da bancada virtual do *Tinkercad* vazia e com a opção de escolha de componentes no menu à direita

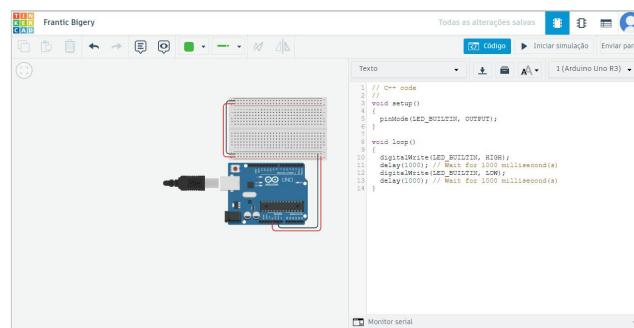
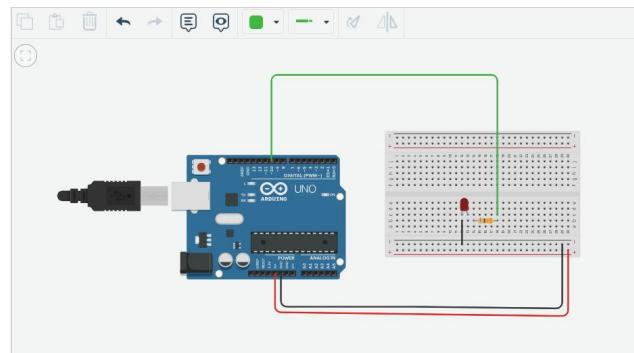
Para iniciar seus experimentos, indicamos esta opção de ferramental pois as simulações não necessitam de dispositivos físicos, fornecendo assim liberdade e autonomia ao desenvolvedor para o teste de circuitos que podem ir de simples a complexos, e os quais podem ser reproduzidos e testados virtualmente.

Como dito anteriormente, no *Tinkercad* podemos utilizar diferentes placas da plataforma *Arduino*, sendo a mesma bastante indicada por ser um produto altamente especializado e *Open Source*. O hardware *Arduino* possui baixo custo e a sua aplicação é bastante simples.

A linguagem de programação utilizada na plataforma *Arduino* é a "C++" ou "C plus plus", porém, adicionando as bibliotecas *Python* corretas às necessidades a serem tratadas, ganhasse bastante agilidade devido à grande disponibilidade de bibliotecas com recursos específicos disponíveis, além do que *Python* por sua estrutura e sintaxe simplificada propicia maior facilidade de programação.

Agora que esclarecemos como se dá a comunicação entre as duas plataformas *Arduino/C++* e *Python*, podemos partir para o exemplo prático, o qual terá como objetivo o envio dos comandos *Python* para a placa *Arduino* por comunicação serial. A integração entre as plataformas e linguagens torna-se possível, com as funções da biblioteca *Python* gerando os códigos necessários para o acesso e interação com os diferentes dispositivos.

Vamos observar então o ato de acender um LED, via uma conexão serial USB - *Universal Serial Bus* ou Porta Serial Universal, ou seja, a *Arduino* atua fisicamente ou no nosso caso através de simulação nas ações de dispositivos e sensores.

**Figura 12.** Demonstração da bancada virtual Autodesk *Tinkercad* com código em C++ e o experimento sendo implementado**Figura 13.** Exemplificação do circuito finalizado no *Tinkercad*

Como o nosso exemplo foi desenvolvido e executado no Sistema Operacional (SO) *MS-Windows*, as bibliotecas *Python* necessárias para comunicação serial entre as duas plataformas tiveram que ser previamente instaladas, para tal utilizamos o gerenciador de pacotes *pip* (*Python Package Index*), através das respectivas linhas de comando descritas a seguir:

pip install pyfirmata // instala a biblioteca pyfirmata escrita em Python

e

pip install pyserial // instala a biblioteca pyserial escrita em Python

Ambas as bibliotecas podem ser encontradas através do próprio sistema de pesquisas da linguagem *Python*, que no nosso caso foi realizado através do IDE - *Integrated Development Environment* ou Ambiente de Desenvolvimento Integrado, *PyCharm*

¹ <<https://www.jetbrains.com/pycharm/>>. O ambiente *PyCharm* apesar de pago, pode ser utilizado gratuitamente para uso acadêmico individual de alunos e professores, para uso em salas de aula, para uso em bootcamps e cursos. Podendo nesta forma de licenciamento ser baixado através do link a seguir: <<https://www.jetbrains.com/community/education/#students/>>

Atenção

Caso seja utilizado o microcontrolador *Raspberry Pi* ao invés do *Arduino*, segue indicação de duas das mais populares bibliotecas necessárias para a interação via Python - *Raspberry Pi*, quais sejam:

- **RPi.GPIO** : Biblioteca Python que permite controlar os pinos GPIO - General Purpose Input/Output do *Raspberry Pi*. Ela é frequentemente utilizada para controlar LEDs, motores, sensores e outros dispositivos conectados aos pinos GPIO. A mesma está disponível para download através do link : <<https://github.com/raspberrypi/gpiozero>>; e
- **gpiozero** : É um módulo Python de alto nível que simplifica o controle de componentes eletrônicos no *Raspberry Pi*, como LEDs, botões, servos, sensores e muito mais. Ele é projetado para ser fácil de usar, especialmente para iniciantes. A mesma está disponível para download através do link : <<https://gpiozero.readthedocs.io/en/stable/>>.

O ambiente de simulação para *Raspberry Pi* mais indicado, é o ***Raspberry Pi Simulator*** presente no *Microsoft Azure IoT*. Ele permite que você desenvolva e teste códigos para o *microcontrolador* em um ambiente virtual antes de implementá-lo no *hardware real* (como no *Tinkercad*).

Você pode acessar o simulador e obter mais informações através dos links :

<<https://azure.microsoft.com/en-us/products/iot-hub/>> ou <<https://azure.microsoft.com/en-us/free/iot/>>

Obs.: Há um período para experimentação gratuita, porém, com tempo limitado.

As bibliotecas e os ambientes de simulação podem variar dependendo do que você deseja realizar, isso para qualquer plataforma.

Dando continuidade ao experimento, vamos então executar as ações para a instalação das bibliotecas necessárias para a interação entre Python e *Arduino*. Para adicionar uma nova biblioteca pelo PyCharm, proceda com os seguintes passos:

Main Menu ou (Alt + \)

File - Settings ou (Ctrl + Alt + S)

Python Interpreter

Install ou (Alt + Insert)

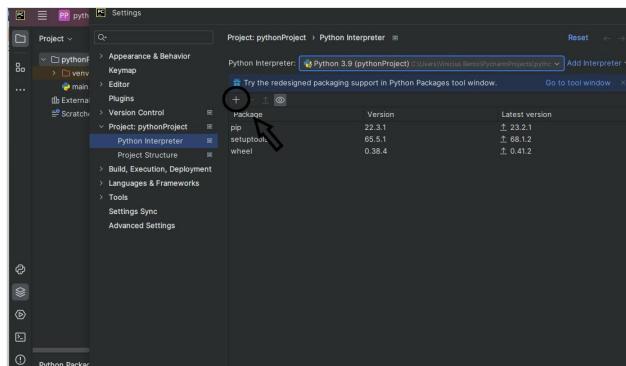


Figura 14. Demonstração do acesso ao local de instalação de bibliotecas/pacotes no PyCharm

Acessando o local (diretório/pasta) desejado, procure as bibliotecas necessárias e pronto, a comunicação via Serial-Serial entre as plataformas Python e *Arduino* poderá ser realizada.

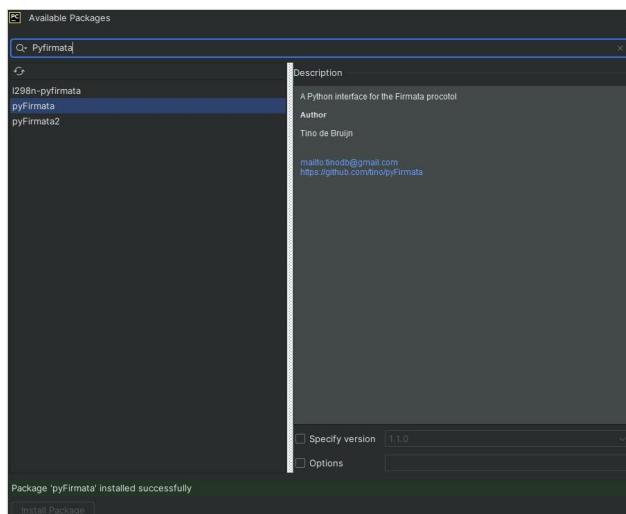
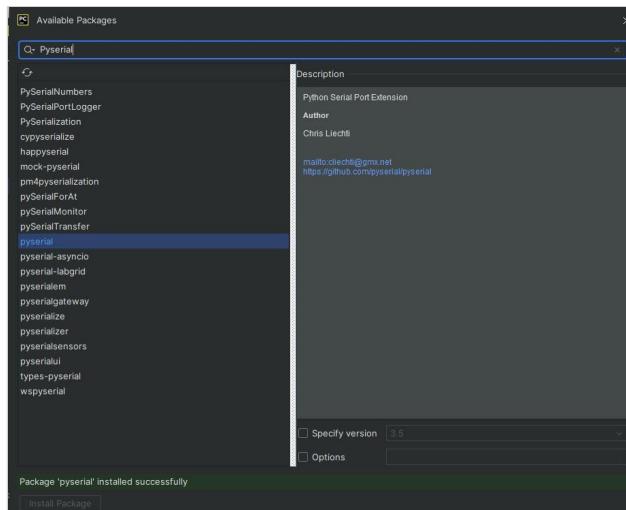


Figura 15. Demonstração do local de instalação da biblioteca *pyFirmata***Figura 16.** Demonstração do local de instalação da função *pyserial***Dica**

Como a comunicação entre as plataformas será realizada via Serial (USB), garanta que a COM (Porta de Comunicação) solicitada está correta. A solução para isso é que sempre seja utilizada a mesma Serial (USB) em aplicações deste tipo.

Experimento: Seguindo com a experiência, instaladas as bibliotecas, iremos executar os códigos de programação em *Python* no ecândio do *PyCharm*.

Python - PyCharm: Código para verificação serial.

```

1 import serial
2 conexao=""
3 for porta in range(10):
4     try:
5         conexao = serial.Serial("COM"+str(porta), 115200, timeout=0.5)
6         print("Conectado na porta: ", conexao.portstr)
7         break
8     except serial.SerialException:
9         pass
10 if conexao!="":
11     conexao.close()
12     print("Conexão encerrada")
13 else:
14     print("Sem portas disponíveis")

```

Figura 17. Demonstração da execução do código do programa [**main.py**] no *PyCharm*

Este código *Python* utiliza a biblioteca **serial** para tentar estabelecer uma conexão serial com um dispositivo em uma das portas disponíveis de um computador ou máquina/dispositivo.

É um IDE desenvolvido pela empresa tcheca *JetBrains*, o qual propicia a facilitação e otimização de programação em linguagem *Python*, fornecendo análise de código, depurador gráfico, testador de unidade integrado, integração com sistemas de controle de versão e que suporta desenvolvimento Web com *Django*.

Descrição detalhada do código:

1. Importa a biblioteca **serial** a qual permitirá a comunicação serial

2. Inicializa uma variável vazia chamada "**conexao**" para armazenar a conexão serial se a mesma for bem-sucedida. Inicia um loop "for" que itera de **0** a **9**, representando as portas COM de **0** a **9** (**COM0**, **COM1**, **COM2**,... até a **COM9**).
 3. No loop, tenta abrir uma conexão serial com a porta COM atual usando a função "**serial.Serial**".
- Os argumentos passados para esta função são: O nome da porta **COM**, construído com base no número da porta atual. A taxa transmissão (*baud rate*) é definida como **115200 bauds**¹. Um tempo limite (*timeout*) de **0.5** segundos para tentar estabelecer a conexão.
4. Se a conexão for estabelecida com sucesso (ou seja, nenhum erro ocorrer), exibe a mensagem "**Conectado na porta:**" informando em qual porta COM a conexão foi estabelecida e encerra o loop usando "**break**".
 5. Se ocorrer uma exceção do tipo "**serial.SerialException**", isso significa que a conexão não pôde ser estabelecida na porta atual. O loop continua para a próxima porta.
 6. Após o loop, verifica se a variável "**conexao**" não está vazia. Se não estiver vazia, significa que uma conexão bem-sucedida foi estabelecida em uma das portas COM. Nesse caso, fecha a conexão e exibe uma mensagem de que a conexão foi encerrada.
 7. Se a variável "**conexao**" ainda estiver vazia após o loop, isso indica que nenhuma das portas COM estava disponível para conexão e exibe a mensagem "**Conexão encerrada**" informando que não há portas disponíveis.

O objetivo principal deste código é encontrar uma porta COM disponível e estabelecer uma conexão serial com a mesma. Ele testa várias portas até encontrar uma que funcione ou até esgotar todas as portas disponíveis.

¹*bauds* - *baud rate* ou “taxa de *baud*”, significa que no nosso exemplo o canal de comunicação pode mudar o estado até 115200 vezes por segundo. O termo “mudar” significa que ele pode variar de 0 para 1 ou de 1 para 0 até X vezes por segundo. Isto se refere também ao estado atual da conexão, como voltagem, frequência ou risco.

Após a verificação das portas COM, podemos a seguir executar o código de interação com os dispositivos IoT:

```

1 import serial
2 conexao=""
3 for porta in range(10):
4     try:
5         conexao = serial.Serial("COM"+str(porta), 115200, timeout=0.5)
6         print("Conectado na porta: ", conexao.portstr)
7         break
8     except serial.SerialException:
9         pass
10 if conexao!="":
11    acao=input("Digite:
12 <L> para Ligar
13 <D> para Desligar: ").upper()
14     while acao=="L" or acao=="D":
15         if acao=="L":
16             conexao.write(b'1')
17         else:
18             conexao.write(b'0')
19     <L> para Ligar
20 <D> para Desligar: ").upper()
21     conexao.close()
22     print("Conexao encerrada")
23 else:
24     print("Sem portas disponiveis")
25

```

Figura 18. Demonstração da execução do código no PyCharm que será enviado ao Arduino

Esse código em *Python* é um programa que tenta estabelecer uma conexão serial com um dispositivo em uma das portas COM disponíveis em um computador e, se bem-sucedido, permite que o usuário envie comandos para ligar ou desligar um dispositivo controlado pela porta serial. Vamos explicar o funcionamento passo a passo, a seguir:

1. A biblioteca **serial** é importada para lidar com a comunicação serial.
2. Inicializa uma variável vazia chamada "**conexao**" para armazenar a conexão serial, se for bem-sucedida.
3. Inicia um loop "for" que itera de **0** a **9**, representando as portas COM de **0** a **9** (**COM0**, **COM1**, **COM2**,... até a **COM9**).
4. No loop, tenta abrir uma conexão serial com a porta COM atual usando **serial.Serial("COM"+str(porta), 115200, timeout=0.5)**. Tenta estabelecer uma conexão com uma velocidade de transmissão de **115200bauds** e um tempo limite de **0.5** segundos para estabelecer a mesma.
5. Se a conexão for estabelecida com sucesso (ou seja, nenhum erro ocorrer), ele exibe a mensagem "**Conectado na porta:**" informando em qual porta COM a conexão foi estabelecida e sai do loop usando "**break**".
6. Se ocorrer uma exceção do tipo "**serial.SerialException**", isso significa que a conexão não pôde ser estabelecida na porta atual. O loop continua para a próxima porta.

7. Após o loop, verifica se a variável "**conexao**" não está vazia. Se não estiver vazia, significa que uma conexão bem-sucedida foi estabelecida com uma das portas COM. Nesse caso, o programa pede ao usuário para digitar "**L**" para ligar ou "**D**" para desligar dispositivo controlado pela porta serial.
8. Enquanto o usuário digitar "**L**" ou "**D**", o programa enviará os comandos correspondentes ("**1**" para ligar e "**0**" para desligar) pelo serial usando **conexao.write(b'1')** ou **conexao.write(b'0')**.
9. Quando o usuário não digitar "**L**" nem "**D**", o programa encerrará a conexão serial, e exibirá a mensagem "**Conexão encerrada**" terminará a execução.
10. Se a variável "**conexao**" ainda estiver vazia após o loop, isso indica que nenhuma das portas COM estava disponível para conexão e o programa exibirá então a mensagem "**Sem portas disponíveis**".

Agora que desenvolvemos todas as funções de acionamento do dispositivo (LED) em *Python*, faremos a integração das duas plataformas transmitindo o código para placa *Arduino* via COM. Para a tal integração utilizaremos o código escrito no *PyCharm*, o qual será enviado para o *Arduino Sketch*, que fará a introdução do código *Python* ao dispositivo conectado à placa através de outro código em linguagem C++. O código C++ que faz parte da integração em conjunto com o código *Python* está ilustrado na Figura 19 a seguir.

Dica

Você não precisa obrigatoriamente instalar o IDE *PyCharm* para programar em *Python* e desenvolver seus projetos como fizemos aqui, existem vários outros pacotes de ferramentas alternativas muito boas, entre elas destacamos:

O *Jupyter Notebook* <<https://jupyter.org/>>, o qual permite criar e compartilhar arquivos interativos chamados de "cadernos" ou *notebooks*. Os cadernos combinam elementos de código, permitindo que os usuários criem e compartilhem relatórios, visualizações de dados e análises exploratórias.

Você também pode utilizar um dos pacotes mais completos de ferramentas *Python*, em especial para análise de dados, IA (Inteligência Artificial) e que traz incorporado além de várias bibliotecas e ferramentas também o *Jupyter*, que é o pacote *Anaconda* <<https://www.anaconda.com/>>, porém para a instalação completa e posterior execução de seus programas com ele mesmo, seu equipamento terá que ser razoavelmente robusto.

Uma opção para equipamentos mais antigos e com menor capacidade de processamento é o *Google Colab "Colaboratory"* <<https://colab.research.google.com/notebook>>, que é uma solução *online* executada diretamente via *web browser*, qual possibilita escrever e executar códigos *Python* sem nenhuma configuração, permite acesso a GPUs - *Graphics Processing Unit* ou Unidades de Processamento Gráfico sem custo financeiro e disponibiliza alta facilidade de compartilhamento de arquivos através do *Google Drive*.

Outra opção é o *IDLE*¹, que apesar de razoavelmente simples é bastante útil para usuários iniciantes, o editor não faz parte da linguagem *Python* mas é lançado conjuntamente em seu pacote a cada liberação da mesma, desde a sua versão 2.3.

Nota - Em algumas distribuições Linux, é necessário instalar o pacote *Python* e o editor *IDLE* separadamente, diferente de como ocorre nos SOs MS-Windows e Mac OS - nestes casos os comandos para as últimas versões disponíveis quando do desenvolvimento de nosso curso, seriam respectivamente:

sudo apt-get install python311 e ***sudo apt-get install idle*** ou ***sudo apt-get install idle3***

¹O *IDLE* - Integrated Development and Learning Environment ou Ambiente Integrado de Desenvolvimento e Aprendizagem, foi completamente escrito em *Python* com o toolkit GUI *Tkinter* da linguagem.

Arduino IDE - *Sketch*¹

O *Arduino Sketch* <<https://www.arduino.cc/en/software>> é um software de execução de códigos em C++ com duas funções principais simples, quais sejam: **void setup()** e **void loop()**



```

sketch_sep05a_ - Experimento_ligar_Led_por_acionamento...
Arquivo Editar Sketch Ferramentas Ajuda
void setup() {
  pinMode(10, OUTPUT);
  Serial.begin(115200);
}

void loop() {
  int valor_recebido;
  if (Serial.available()){
    valor_recebido = Serial.read();
    if (valor_recebido=='0'){
      digitalWrite(10, LOW);
    }else{
      digitalWrite(10, HIGH);
    }
  }
}

Compilação terminada.
O sketch usa 1750 bytes (5%) de espaço de armazenamento para programa
Variáveis globais usam 184 bytes (8%) de memória dinâmica, deixando
13 bytes para variáveis locais.

```

Figura 19. Demonstração da execução do código em *Arduino*

Agora no *Arduino Sketch*, vamos determinar os parâmetros de interação do código com os dispositivos (no nosso caso, o LED) inseridos no circuito através de dois métodos:

- **void setup()** - É o método que determina as execuções dos códigos e parâmetros ao ligar o *Arduino*, a função **setup()** é chamada uma vez quando o microcontrolador é inicializado, sendo utilizada para configurar o ambiente do *Arduino* antes de começar a executar o código principal. Nela são configurados os pinos, definida a taxa de comunicação serial (se necessário). Por exemplo, você estiver usando sensores ou atuadores, é na função **setup()** que você definirá como esses componentes serão utilizados e conectados; e
- **void loop()** - Este método realiza sequências infinitas ou pré-determinadas dos parâmetros e códigos configurados, com a função **loop()** sendo executada continuamente após o **setup()**, e sendo responsável por controlar o comportamento do *Arduino* enquanto ele estiver em execução. Ou seja, com a função **loop()** o código *Arduino* é executado repetidamente.

Explicando o código

Setup (Configuração): Define o **pino 10** como uma saída (**OUTPUT**) - sendo o mesmo identificado por um fio verde conectando placa de testes da bancada virtual do *Thinkercad*, como detalhado no diagrama da Figura 13 - o que significa que ele será usado para enviar sinais de saída. Inicializa a comunicação serial com uma taxa de **115200 bauds**.

Loop (Laço): Declara uma variável chamada **valor_recebido** para armazenar os dados recebidos pela porta serial. Verifica se há dados disponíveis na porta usando **Serial.available()**. Isso verifica se algo foi enviado para o *Arduino* via comunicação serial. Se houver dados disponíveis, lê o valor recebido usando **Serial.read()** e armazena na variável **valor_recebido**. Em seguida, verifica se o valor recebido é igual a '0'; caso seja, ele define o **pino 10** (que foi configurado como saída) como **LOW** (desligado) ou define o **pino 10** como **HIGH** (ligado).

Em resumo, este código configura o **pino 10** como um pino de saída e lê os dados recebidos na porta serial. Basicamente, ele configura o estado do **pino 10** com base no valor recebido pela porta serial, agindo como um interruptor ligar/desligar.

Concluindo, o nosso exemplo demonstrou como o *Arduino* em conjunto com código em linguagem *Python* pode estabelecer uma conexão serial com uma porta COM, de modo a permitir ao usuário o controle de um dispositivo ligando-o ao digitar "L" ou desligando-o digitando "D" através de uma porta serial.

Notas:

Para que o experimento de comunicação seja completo, será necessário ter disponível uma placa física *Arduino*, pois não é possível executar a comunicação serial pelo *Tinkercad*.

Isso se dá pelo fato de não ser possível definir qual porta COM poderia ser conectada em uma placa simulada, todavia o exemplo demonstra a funcionalidade que pode ser aplicada para exemplos reais. De qualquer forma ao observar com atenção as informações aplicadas neste experimento simples, ficam claras as possibilidades de criação de outros experimentos integrando *Python* e *Arduino* utilizando dispositivos IoT.

Dicas:

Indicamos a seguir, alguns exemplos práticos em vídeo disponíveis no *YouTube*, os quais irão propiciar uma melhor assimilação do que tratamos até aqui. Neles demonstram-se possíveis funções para o funcionamento da IoT, utilizando a linguagem *Python* e as plataformas *Arduino* e *Python*.

- Como acender a LUZ do seu quarto com ARDUINO e PYTHON!?

<https://www.youtube.com/watch?v=2bDmjUxCU0M>
- Monitor de temperatura e umidade com ESP8266 e interface gráfica em python

<https://www.youtube.com/watch?v=45VCTB7f0sM>
- APRENDA a dar os PRIMEIROS PASSOS em Pythonna sua Raspberry Pi Pico

<https://www.youtube.com/watch?v=suFAltbpNBM>
- ArduinoxRaspberry Pi: Qual o Melhor? com Python e Pyfirmata- Desafio Maker

<https://www.youtube.com/watch?v=q7u5KUM94Uc>

¹Sketch é simplesmente o termo utilizado para descrever o código que é escrito para programar o microcontrolador Arduino.

Arquitetura de Sensores IoT

A arquitetura da IoT é caracterizada por ser específica e dividida em camadas, sendo que essas podem ser alteradas conforme a funções aplicadas ao sistema, e mesmo que elas mudem em função das necessidades, todas derivam das seguintes: **camada de percepção**; **camada de rede**; e **camada de aplicação**. Logo, tal arquitetura descreve a interação entre os componentes do sistema permitindo assim representar de forma detalhada os componentes de *hardware*, os **protocolos**¹ de comunicação e a interação deste o software e como os dispositivos irão interagir com o cliente ou “quem irá usufruir da arquitetura” em sua totalidade.

De forma mais específica, temos:

Na **camada de aplicação** os programas, geralmente de uma interface robusta, a qual interage com o usuário ao final de cada aplicação;

Na **camada de rede** os dispositivos encarregados de gerenciar o tráfego de informação através de uma rede, usando protocolos específicos, tais como: HTTP, HTTPS, Bluetooth, Zigbee, NFC, etc.; e

E por fim, a **camada de percepção** que se caracteriza como a camada física, onde estão as “coisas”.

Essas camadas são fundamentais para outras arquiteturas existirem, observando sua organização heterogenia com as suas várias funções aplicadas a IoT, é possível perceber que dados serão obtidos de todo ecossistema, sendo que os mesmos dentro de uma vi de **negócios** significam a “capitalização de recursos” - método bastante utilizado por exemplo pelas empresas Big Techs detentoras redes sociais para tornarem mais assertivas as publicidades divulgadas em suas plataformas - o que comercialmente é imprescindível para a sustentabilidade de qualquer projeto ou ação, ou seja, dessa capitalização emerge uma “**camada de negócios**” a qual apesar não ser fisicamente presente, converge das três camadas básicas da arquitetura.

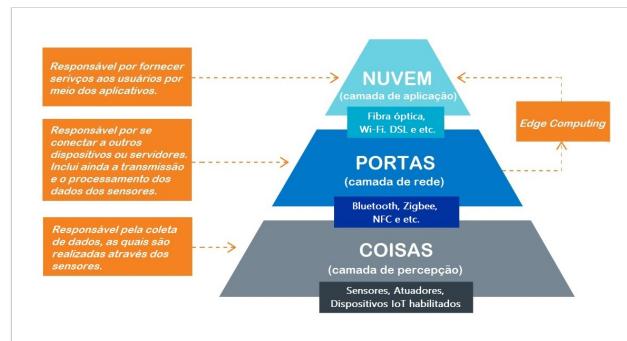


Figura 20. Camadas fundamentais da arquitetura IoT

¹ Protocolos consistem de conjuntos de regras que regem o formato dos dados comunicados pelas redes em geral e pela Internet, ou seja, são as regras “sintaxe, se e sincronização” que permitem que a comunicação entre computadores conectados ocorra de forma organizada e segura.

Computação em Nuvem (*Cloud Computing*)

Cloud Computing são redes de servidores remotos de aplicações, hospedados na Internet para armazenamento e recuperação de dados. As "nuvens" fornecem uma série de serviços de TI, tais como servidores, bancos de dados, softwares, armazenamento virtual entre outros.

A computação em nuvem é alicerçada em uma infraestrutura virtual, a qual permite armazenar e acessar dados e serviços praticamente sem limitações. Esse tipo de arquitetura consiste em entregar soluções, como serviços de computação, disponibilizadas aplicativos e armazenamento de dados via Internet, sem a necessidade desses mesmos serviços estarem dispostos fisicamente.

Os modelos de serviço em nuvem via-de-regra são disponibilizados por plataformas de grandes empresas da área de TI, as quais fornecem de modo "à la carte" aos seus usuários, opções de escolha específicas dos recursos que serão alocados em cada projeto. Esse modelo favorece entre outras coisas, a possibilidade de escala e/ou elasticidade, sendo os serviços/recursos pagos/bilhetados (quais são) apenas pelo o que se utiliza em referência à armazenagem e outros recursos tais como (processadores, SOs, bancos de dados, links, etc.).

Pelo porte das infraestruturas disponibilizadas pelas plataformas especializadas em oferecer serviços em nuvem, a segurança física dos dados armazenados é maior que a convencional, pois são disponibilizadas redundância de hardware, segurança lógica e recursos de monitoramento 24x7x365, atualizações contínuas de software e de regras de uso (boas práticas), inclusive com rápido atendimento às legislações impostas em diferentes regiões do globo.

Exemplos cotidianos da utilização massiva de *Cloud Computing*, são os sensores que se conectam por meio de dispositivos concentradores como os celulares, os quais trocam dados através de nuvens que permitem processamento em tempo real destes dados, com os mesmos sendo acessados por seus usuários através de interfaces de aplicação (APIs), seja em servidores ou qualquer outro dispositivo de sistema em tempo real. Essas ações seriam impossíveis caso não houvessem aplicações em nuvem suportando telefones celulares, qualquer outro dispositivo com possibilidade de acesso à Internet e capacidade de recepção e transmissão de dados.

Dispositivos Físicos (Coisas)

São todos os dispositivos físicos capazes de armazenar e transmitir informações dentro do sistema implementado, tais como: celulares, computadores, sensores, atuadores, gateways, dentre outros.

Comunicação e Protocolos de Rede para IoT

Atualmente os equipamentos disponíveis no mercado para utilização em projetos IoT se comunicam entre outras, através das seguintes tecnologias de transmissão: *Bluetooth*, NFC, 4G, 5G e *Wi-Fi*. Tais tecnologias exercem a função de facilitar e garantirem o acesso aos dados, os quais ao serem processados irão gerar informações que futuramente auxiliarão no controle do sistema implementado.

As mesmas precisam ser avaliadas no processo de análise sob a visão da **arquitetura do sistema**, pois por ser um sistema heterogêneo e com muitas possibilidades de aplicação, é necessário avaliar como os dispositivos de comunicação se conectarão e quanto irão consumir energeticamente, sendo tal análise realizada com base na infraestrutura disponível. Portanto, o objetivo comum para essas tecnologias é o de obter uma comunicação de longo alcance, alta velocidade e baixo consumo.

Conceito de Plataforma IoT

As plataformas utilizadas para aplicações em IoT necessitam centralizar diversas funções em um ambiente conectado, as mesmas podem ser as mais diversas, tais como: acender luzes, acionar aparelhos de ar-condicionado, ligar/desligar televisores, entre outros. As plataformas que fornecem tais funcionalidades "serviços" são identificadas como: **IaaS - Infrastructure as a Service** ou Infraestrutura como Serviço; **PaaS - Platform as a Service** ou Plataforma como Serviço e **SaaS - Software as a Service** ou Software como Serviço.

Na IoT assim como em outros contextos de computação em nuvem, os modelos de serviço **IaaS**, **PaaS** e **SaaS** podem ser aplicados, embora suas funções sejam ligeiramente diferentes como veremos a seguir.

IaaS fornecem infraestrutura de hardware virtualizada, como servidores, armazenamento e rede, que pode ser usada para criar e gerenciar implantações de IoT. As plataformas IaaS, são bastante úteis quando se necessita de controle granular "fragmentado" sobre a infraestrutura, pois permitem criar sua própria arquitetura de IoT personalizada, o que é comum em cenários em que a conectividade direta com dispositivos ou gateways é essencial, todavia ao contrário das PaaS, elas não incluem um ambiente de desenvolvimento específico.

PaaS são plataformas de serviços em nuvem que fornecem infraestrutura e ambiente de desenvolvimento específicos para a criação de aplicativos de IoT. Elas oferecem uma plataforma completa que inclui recursos de conectividade, armazenamento de dados, processamento e gerenciamento de dispositivos IoT.

São adequadas para desenvolvedores que desejam criar aplicativos IoT personalizados sem se preocuparem com a complexidade da infraestrutura subjacente. Elas fornecem também ferramentas para coletar, armazenar e analisar dados de dispositivos IoT, o que simplifica o desenvolvimento de aplicativos que se comunicam com os mesmos.

SaaS referem-se a aplicativos de IoT que são entregues como serviços prontos para uso. Isso pode incluir aplicativos de monitoramento, rastreamento ou gerenciamento de dispositivos. São mais adequadas para usuários finais ou empresas que desejam acessar funcionalidades de IoT sem a necessidade de desenvolver seus próprios aplicativos ou gerenciar a infraestrutura, neste fornecendo os usuários simplesmente se inscrevem para usar o serviço e aproveitam as funcionalidades fornecidas.

As soluções em plataforma podem ser utilizadas em diversas situações, com os modelos de serviço em nuvem possuindo objetivos ligeiramente diferentes, abstraindo a complexidade do trabalho de seus usuários “clientes” em ordem crescente em razão das funcionalidades oferecidas em cada modelo. Sendo a escolha entre os mesmos, dependente dos requisitos específicos do projeto de IoT em questão e da experiência técnica das equipes que os executarão e dos objetivos que se desejam alcançar.

Diversas empresas exploram o ramo da IoT, as quais oferecem diferentes plataformas com recursos e estruturas customizadas. Iremos seguir apresentando algumas opções disponíveis no mercado, as quais oferecem os tipos de infraestrutura descritas.

Plataformas IaaS

Microsoft Azure

É uma plataforma de nuvem da Microsoft <<https://azure.microsoft.com/pt-br/>> a qual oferece uma variedade de serviços IaaS e PaaS. Ela inclui máquinas virtuais, armazenamento, bancos de dados, entre outros recursos.

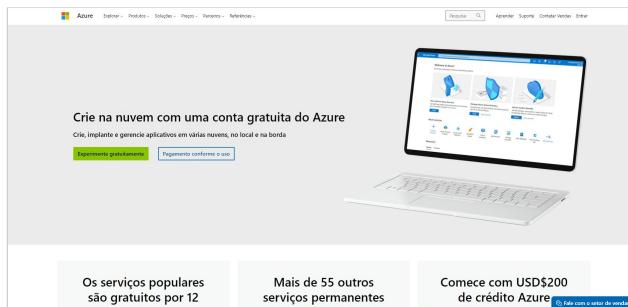


Figura 21. Fragmento inicial da página oficial do Microsoft Azure

IBM Cloud

Oferece uma ampla variedade de serviços em nuvem, incluindo máquinas virtuais, armazenamento, serviços de rede e muito mais. A plataforma da IBM <<https://www.ibm.com/cloud>> é reconhecida por seu foco em empresas e soluções empresariais.

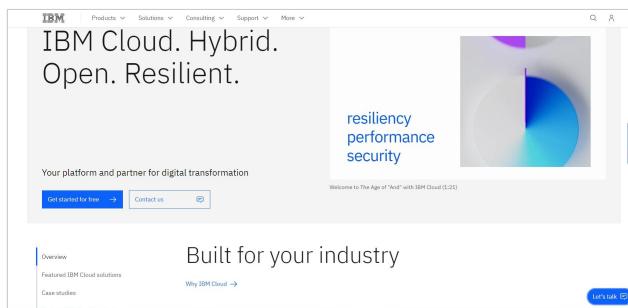


Figura 22. Fragmento inicial da página oficial do IBM Cloud

Amazon Web Services (AWS)

Os serviços AWS - Amazon Web Services <<https://aws.amazon.com/pt/>> oferecem recursos em IaaS, PaaS e SaaS, dentre eles o desenvolvimento de infraestruturas de TI virtualizadas (foco do IaaS), as quais incluem a capacidade de provisionar servidores virtuais, armazenamento em nuvem, serviços em rede e etc. diretamente através do web site da empresa e aplicando-as conforme às suas necessidades.

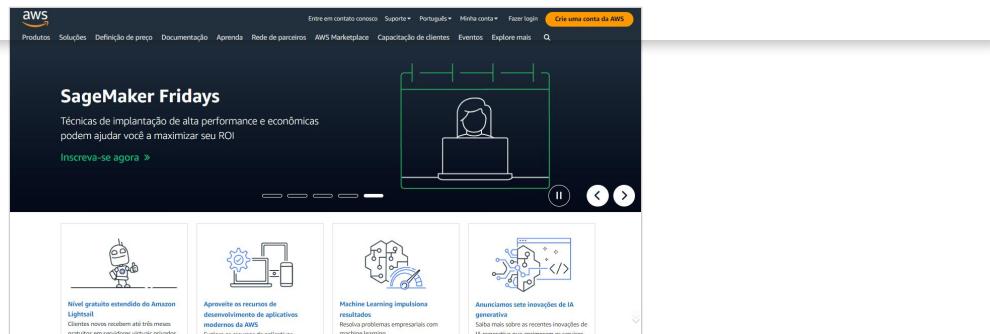


Figura 23. Fragmento inicial da página oficial do AWS da Amazon

PaaS (*Platform as a Service*)

Esse tipo de plataforma de serviços em nuvem fornece infraestrutura e ambiente de desenvolvimento para a criação de aplicativos, incluindo recursos de conectividade, armazenamento de dados, processamento e gerenciamento de dispositivos IoT.

PaaS em IoT é adequada para desenvolvedores que desejam criar aplicativos personalizados sem se preocupar com a complexidade das infraestruturas implícitas, fornecendo ferramentas para coletar, armazenar e analisar dados de dispositivos IoT.

Plataformas PaaS e SaaS

Heroku

Diversas empresas de renome disponibilizam acesso a plataformas PaaS com base em suas infraestruturas, como é o caso da empresa americana Heroku <<https://www.heroku.com/>>, a qual oferece ambiente para desenvolvimento, implementação e gerenciar de aplicativos Web e que é bastante utilizada por desenvolvedores de aplicativos e sistemas.

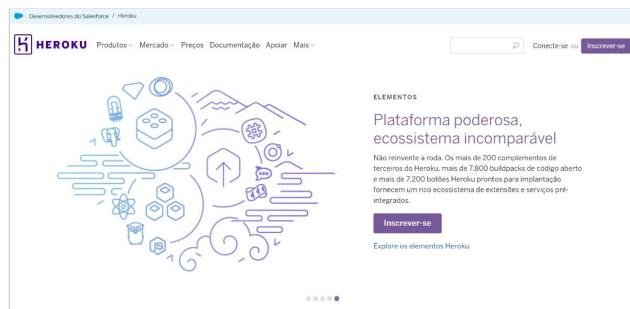


Figura 24. Fragmento inicial da página oficial da Heroku

Existem também empresas que disponibilizam modelos especializados em SaaS, o qual utiliza modelo e princípios de negócio semelhantes ao PaaS, porém de forma mais especializada. São exemplos de empresas que trabalham com esse tipo de tecnologia a **Oracle** e o **Google**.

Oracle - Cloud Platform

O Cloud Platform <<https://www.oracle.com/br/cloud/>> oferece serviços especializados de integração de dados, análise de Business Intelligence (BI), bancos de dados, Big Data, Inteligência Artificial, Aprendizado de Máquina, entre outros.

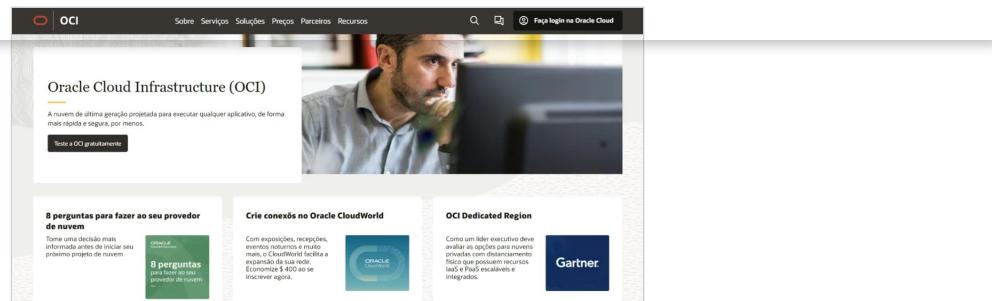


Figura 25. Fragmento inicial da página oficial do *Cloud Platform*

Google Cloud - App Engine

O Google Cloud App Engine <<https://cloud.google.com/appengine>>, fornece um grande conjunto de recursos para o modelo SaaS, dentre os quais suporte a diferentes linguagens de desenvolvimento, tais como (Python, C#, Java, etc.), diagnóstico avançado de aplicativos com captura de erros, controle de versão de aplicativos, ferramentas de segurança entre outras funcionalidades.



Figura 26. Fragmento inicial da página oficial do *Google Cloud*

A ampla variedade de plataformas e equipamentos disponíveis no mercado, é um testemunho real de que existem múltiplas soluções para atender a casos específicos, o que ficou visível nas diferentes opções elencadas anteriormente. Portanto, é de suma importância realizar uma análise criteriosa e viabilizar a melhor relação custo-benefício, optando pelas infraestruturas de armazenagem, redundância, segurança, etc. mais adequadas aos dados a serem tratados.

Entre as razões pelas quais a escolha equilibrada entre custos-benefícios em IoT são fundamentais, citamos as que seguem:

Otimização de Recursos Financeiros: Muitos projetos de IoT operam com orçamentos limitados. Ao escolher uma plataforma que oferece uma relação custo-benefício favorável, você pode maximizar o uso dos recursos financeiros disponíveis, alocando-os de forma eficaz para garantir que as necessidades do projeto sejam atendidas sem exceder o orçamento.

Escalabilidade: À medida que os projetos de IoT crescem e evoluem, a capacidade de escalabilidade da plataforma se torna crucial. Portanto, a escolha adequada garantirá que você possa expandir sua infraestrutura de acordo com as demandas sem incorrer em custos excessivos.

Funcionalidades Essenciais: É importante avaliar se a plataforma oferece as funcionalidades essenciais necessárias à arquitetura do seu projeto. Priorize recursos que agreguem valor real, evitando gastos com recursos excessivamente complexos que não serão utilizados.

Suprimento e Manutenção: Considere de forma criteriosa o suporte técnico e as opções de manutenção oferecidas pela plataforma contratada. Essa escolha deve levar em conta não apenas o custo inicial, mas também os custos contínuos associados ao suporte e manutenção que sempre ocorrerão.

Segurança e Conformidade: A segurança é algo cada vez mais crítica em projetos IoT, especialmente quando dados sensíveis estão envolvidos. Certifique-se de que a plataforma escolhida atenda aos padrões de segurança e conformidade relevantes legais, mantendo seu projeto protegido contra ameaças cibernéticas e cíveis.

Facilidade de Integração: A integração com outros sistemas e dispositivos é fundamental para a grande maioria de projetos IoT. Escolher uma plataforma que facilite integrações pode economizar tempo e recursos significativos.

Análise de Dados: A capacidade de coletar, armazenar e analisar dados é um dos principais benefícios da IoT. Avalie se a plataforma oferece ferramentas adequadas para análise de dados, que permitam a tomada de decisões informadas de forma rápida e eficiente.

A escolha de plataformas que oferecem serviços adequados de infraestrutura é uma decisão crucial para o sucesso de qualquer projeto IoT. Sendo um dos fatores essenciais a serem considerados ao tomar essa decisão, o equilíbrio entre custo e benefício, o que se reflete na avaliação cuidadosa dos custos associados à plataforma em relação aos benefícios que ela proporcionará ao projeto.

Saiba Mais

A seguir indicamos alguns vídeos disponíveis no YouTube, os quais abordam exemplos práticos de integração de dispositivos e sensores IoT com conexão à Internet. Não deixe de assisti-los, pois os mesmos disponibilizam informações importantes sobre o que tratamos até aqui.

- **IoT PARA INICIANTES**

<https://www.youtube.com/watch?v=-ym9zvsQsOo>

- **CONHEÇA 5 PROJETOS IoT INCRÍVEIS - O último eu construí para colocar no meu laboratório.**

<https://www.youtube.com/watch?v=kI2gdbmk9Xw>

- **Plataforma de IoT industrial Interoperável**

<https://www.youtube.com/watch?v=40dt-EGMVR4>

- **Conectando sensores à aplicação final de IoT**

<https://www.youtube.com/watch?v=TMmd0W38qnw>

Integração com serviços Web e APIs externas

As APIs - *Application Programming Interface* ou Interface de Programação de Aplicativos, são conjuntos de serviços/funções intermediárias disponibilizadas para que outros programas/aplicativos possam utilizá-las diretamente e de forma simplificada, ou seja, os desenvolvedores tenham que se envolver necessariamente com detalhes complexos de implementação de códigos de programação.

Certos sites disponibilizam por exemplo, opções de *login* automático através de contas de *e-mail* e isso só é possível, pois entre dois domínios há um acordo de obtenção de informações. Informações essas que estão disponíveis dentro do código de programação APIs, as quais têm com o objetivo automatizar procedimentos e desenvolver novas funcionalidades de forma padronizada, segura e facilitada.

Para que esse método seja corretamente utilizado e obtenha uma melhor integração entre programas ou aplicativos, normalmente utilizadas diferentes tecnologias Web, tais como as que citamos a seguir:

- **XML - Extensible Markup Language:** É uma linguagem de marcação que permite estruturar dados de forma hierárquica usando *tags*¹. É amplamente utilizada em comunicações e trocas de dados entre sistemas e aplicativos. Sua vantagem é ser legível por humanos e amplamente suportada por várias linguagens de programação.
- **YAML - Yet Another Markup Language/Ain't Markup Language:** Este é um formato de serialização de dados que se concentra em ser fácil de ler e escrever para humanos. É frequentemente usado em configurações e arquivos de configuração. É de fácil legibilidade, suportando comentários e por tais razões, largamente utilizado em configurações de infraestrutura.
- **JSON - JavaScript Object Notation:** É um formato de intercâmbio de dados leve e fácil de ler, sendo amplamente utilizado na Web para transmitir dados entre servidores e usuários. Figura como uma escolha comum para as APIs *RESTful*².
- **REST - Representational State Transfer:** Um estilo arquitetural usado para projetar sistemas de comunicação na Web, basear na ideia de que recursos (como objetos de dados) identificados por URLs podem ser manipulados utilizando operações HTTP padrão, como: GET, POST, PUT, PATCH, OPTIONS e DELETE. Por ser simples de entender e usar, são amplamente adotadas na construção de serviços Web de forma escalável.

APIs portanto facilitam aos desenvolvedores criarem aplicativos a serem executados de forma integrada na Web, sendo que a esses entre elas depende do que o projeto necessita, ou seja, que informações você quer obter ou trocar. Por exemplo, de redes sociais, "acesso a informações de perfis" de usuários ou para efetuar buscas mais assertivas sobre o "ranqueamento de diferentes produtos" na Internet, como as que empresas Facebook e Google disponibilizam através de suas APIs.

Existem diferentes tipos de APIs, as quais dependem de modelos pré-estabelecidos de compartilhamento de dados, e portanto, com modos de acesso e/ou formas de uso particulares.

APIs - Políticas de Uso

- APIs PÚBLICAS E ABERTAS (dados abertos a usuários e desenvolvedores, podendo ser utilizadas com algumas restrições por qualquer pessoa);
- APIs PRIVADAS OU INTERNAS (estão sob domínio interno da organização);
- APIs DE PARCEIROS DE NEGÓCIO (dados expostos mediante a acordo comercial); e
- APIs COMPOSTAS (os dados são fornecidos com regras e políticas mistas, de acordo com o desejado e acordado entre as partes)

APIs - Caso de Uso

- API de dados (permitem comunicação entre uma aplicação e um sistema de gestão de base de dados, utilizando **CRUD³**, bancos de dados e provedores SaaS);
- APIs de sistemas operacionais (utilizam dados e recursos disponíveis no sistema operacional);
- APIs Remotas (são externas ao computador que realiza solicitações); e
- APIs Web (as mais utilizadas são as baseadas na *Web* ou no modelo cliente-servidor).

As APIs simplificam a maneira como os desenvolvedores integram novos componentes de aplicações, por isso ajudam na colaboração entre as empresas e as equipes de TI, conectando soluções e serviços, sem a necessidade de se saber como elementos cruciais foram implementados. APIs oferecem, portanto, a flexibilidade necessária para simplificar o design, a administração e o uso de recursos muitas vezes novos e extremamente complexos.

Por suas características, APIs trazem ótimas oportunidades de inovação com esforço e custos reduzidos, o que para a área de IC se transforma constantemente, pode significar a possibilidade de redefinições rápidas de percursos e/ou mudanças tecnológicas que façam frente eficientemente a novos concorrentes.

Dica

Sugerimos que você assista aos dois vídeos a seguir (se possível também a outros), os quais lhe trarão uma visão mais clara “mão na massa”, sobre as facilidades proporcionadas pelas APIs, bem como formas de utilizá-las corretamente e eficazmente.

- **O que é API (*Application Programming Interface*) e Como Usar API com Python?**

<https://www.youtube.com/watch?v=ee1OVldfUw&t=3s>

- **Aprenda a Usar APIs // Mão no Código #7**

<https://www.youtube.com/watch?v=lc0VOosnIAc>

¹ Tags funcionam como etiquetas, as quais etiquetam “tagueam” as buscas por eixos temáticos em sites, elas basicamente são de três tipos: Estruturais (mudam o posicionamento de elementos na *Web*), Cabeçalho (relacionadas com o cabeçalho e alguns elementos básicos de sites “como título e favicon” e de Links (utilizadas para gerenciar qualquer tipo de link existente em páginas *Web*)

² Para ser uma **API RESTful** é necessário que a API se adeque aos 4 níveis de conformidade da arquitetura REST, o que não é obrigatório, mas que geralmente está presente em APIs disponibilizadas por Big Techs e/ou que obedecem a padrões rígidos para a troca de dados.

³ CRUD (Create, Read, Update, Delete- Criar, Ler, Alterar, Apagar) é um acrônimo para as maneiras de se operar em informações armazenadas. É um mnemônico para quatro operações básicas de armazenamento persistente, ou conjunto de ações que podem ser realizadas em “bancos/bases de dados”.

Conceitos de Segurança em IoT

Com o avanço tecnológico equipamentos e processos estão em constante transformação, consequentemente as formas de se proteger dispositivos e informações precisam acompanhar tais mudanças. A segurança é um ponto crucial para sistemas telemáticos e para a privacidade, isso não é diferente, pois todo dado gerado dentro de seu ecossistema pode ser uma arma nas mãos de **crackers¹** e pessoas mal-intencionadas.

À medida que mais e mais dispositivos conectados são incorporados em nossas vidas, cresce também a necessidade de se proteger as infraestruturas que os suportam contra ameaças cibernéticas. E o ecossistema IoT precisa estar protegido por completo, desde os **dispositivos** instalados à **comunicação** entre os mesmos, passando pela **proteção** e o **sigilo** das **informações** geradas pelo sistema.

A seguir elencamos algumas características da IoT, que dificultam a segurança em seus ecossistemas:

- **Diversidade de Dispositivos:** Sistemas IoT abrangem uma ampla gama de dispositivos, desde sensores simples até dispositivos de consumo complexos. Essa diversidade torna difícil aplicar medidas de segurança uniformes e consistentes em todos os dispositivos.
- **Escalabilidade:** A natureza escalável dos sistemas IoT implica que eles podem rapidamente se expandir para incluir milhares e milhões de dispositivos. Gerenciar e proteger essa escala requer soluções robustas e abrangentes de segurança.
- **Conectividade Prolongada:** Muitos dispositivos IoT permanecem conectados continuamente à Internet, o que aumenta sua exposição a ameaças cibernéticas, sendo necessário manter esses dispositivos protegidos por longos períodos.
- **Limitações de Recursos:** Dispositivos IoT frequentemente têm recursos limitados, como poder de processamento, memória e capacidade de armazenamento, portanto implementar medidas de segurança sem afetar o desempenho e a eficiência é um desafio.

- **Padrões e Interoperabilidade:** A falta de padrões de segurança comuns e a interoperabilidade entre dispositivos de diferentes fabricantes pode criar vulnerabilidades, uma vez que apenas um dispositivo inseguro pode comprometer todo o sistema.
- **Coleta e Armazenamento de Dados Sensíveis:** Muitos sistemas IoT lidam com dados sensíveis, como informações pessoais e saúde. Garantir a privacidade e a integridade desses dados é fundamental e complexo.
- **Atualizações e Manutenção:** A aplicação de patches de segurança e atualizações em dispositivos IoT distribuídos em grande escala é desafiadora. Muitos dispositivos podem não ser atualizados regularmente e outros se quer terem mantidas atualizações por seus fabricantes.
- **Ameaças Emergentes:** À medida que a tecnologia evolui, novas ameaças cibernéticas emergem, o que exige uma abordagem proativa para antecipar e mitigar possíveis ameaças.

¹ Diferente do que muitos pensam ao associarem tais comportamentos aos *hackers*, invasões nas áreas ligadas à Internet são via-de-regra, realizadas por **crackers** que são indivíduos que usam suas habilidades em computação para violar a segurança de sistemas e redes de dados.

O Desafio de Gerar Segurança em Sistemas IoT

A maioria dos dispositivos IoT não foram construídos com o intuito principal de gerar segurança, além do que alguns dispositivos de rede não conseguem se quer identificar todos os tipos de dispositivos existentes, o que torna a questão de segurança bastante complexa já que todos os dispositivos estarão interconectados entre si.

Como forma de se organizar esse verdadeiro caos e minimizar os seus inerentes riscos, há a necessidade de se estabelecer um planejamento ou uma ordem de políticas de proteção e utilização dos equipamentos na rede. Tais ações contribuirão também para o entendimento da segmentação das falhas de segurança de cada grupo de dispositivos, com base nisso diferentes perfis de risco serão identificados e poderão ser tratados de forma personalizada.

A seguir elencamos um conjunto de regras “boas práticas”, que embora básicas aumentam em muito a segurança dos sistemas IoT que sejam:

- Manter aplicativos, SOs e softwares em geral sempre atualizados;
- Utilizar senhas fortes e que sejam alteradas de tempos em tempos;
- Sempre que possível manter uma rede exclusiva para seus dispositivos evitando redes abertas e com conexões inseguras;
- Realizar o controle do tráfego de informações entre os dispositivos;
- Garantir a integridade dos dispositivos; e
- Desconectar dispositivos em desuso.

Até aqui avançamos em relação à segurança, porém, ainda é necessário entendermos que os dispositivos IoT necessitam de aplicações de segurança específicas.

Desafios e Riscos de Segurança em IoT

Dispositivos podem sofrer ataques com códigos “mal-intencionados/maliciosos”, os famosos vírus, os quais já podem vir inseridos no dispositivo ou o serem através da rede. Tais códigos são capazes de reprogramar funções, podendo gerar dados incorretos, sequências de informações ou sequestrarem o próprio dispositivo através das funções do equipamento e tomado totalmente o seu controle de forma remota.

Esses mesmos códigos podem comprometer a comunicação e a integridade do sistema ao interceptarem dados por ele trafegados que pode ser ainda potencializado caso as informações capturadas não estejam criptografadas, com o agravante de que mesmo dados criptografados poderão estar sujeitos a procedimentos de **criptoanálise**¹.

Sabendo que os dispositivos em IoT podem apresentar falhas de proteção por conta de seu hardware e desenvolvimento, aliás comum em qualquer sistema comum que possui acesso à Internet, é necessário manter minimamente como rotina as regras básicas listadas anteriormente.

¹ É um método capaz de entender a lógica de criptografia e quebrar as chaves de acesso, deixando os dados coletados e transmitidos desprotegidos.

Técnicas de Criptografia e Autenticação em IoT

Ainda sobre a proteção dos sistemas, um conjunto que pode fortalecer a segurança de dados é a aplicação conjunta de criptografia e autenticação. Mas o que é autenticação? O processo de autenticação fornece garantias sobre a identidade de um usuário, ou seja, é o processo que vai informar ao sistema se você "reclamante" é quem de fato, quem você diz ser!

Chama-se de reclamante a pessoa ou sistema, ou até mesmo o dispositivo de *hardware*, cuja identidade precisa ser verificada e autenticada. Sendo as suas credenciais consideradas as evidências que o reclamante tem que apresentar para estabelecer a sua liberação aos dados requisitados.

Existem alguns tipos de autenticações distintas, como, por exemplo, a autenticação de identidade. Neste caso, quando um usuário afirma ter uma identidade legítima ao sistema, há uma conta no sistema provando a sua autenticidade. Também é possível fazer a autenticação de origem de dados, que é um tipo de autenticação que fornece uma evidência de que os dados (informações, arquivo, uma mensagem de e-mail por exemplo) foram originados de um usuário legítimo. Por exemplo: Você recebe um e-mail de um usuário e aplicando o processo de autenticação de origem de dados, você consegue saber se essa pessoa "X" realmente mandou aquele e-mail ou se o e-mail é forjado.

Independentemente do sistema, sendo ou não computadorizado, o processo de autenticação é de suma importância pois procura por "pistas" de confirmação, o que pode se dar por um crachá, um cartão, uma "chave privada" dentro de um "smart card" ou algum objeto/item físico. Existe ainda o paradigma do algo que você é, nesse caso, o mesmo se baseia em características que são imutáveis no físico do reclamante, como é o caso da autenticidade através de características biométricas.

Os Principais Padrões

Os padrões mais empregados e com maior variação e combinações são as senhas e *PINs - Personal Identification Number*. As senhas são o mecanismo mais comum e mais utilizado para autenticar usuários, porém esse padrão de proteção não é muito confiável, pois representa uma das maiores vulnerabilidades em um sistema de autenticação, isso porque entre outros pontos é dependente de ser "fortes", algo que dificilmente os usuários desenvolvem e/ou aplicam. Senhas também podem ser descobertas facilmente se forem transmitidas sem criptografia pelos sistemas conectados a redes.

Ainda hoje, muitos sistemas não usam criptografia e uma solução para isso pode ser a autenticação de fator duplo, utilizando token, biometria, senha + palavra-chave, entre outras. Além disso, a autenticação em dispositivos pode seguir um padrão lógico estabelecido pelo servidor ao qual estão conectados, com autenticações do tipo:

- **Autenticação Unilateral:** Quando somente um dispositivo necessita demonstrar seus certificados, os quais serão analisados pelo outro dispositivo já conectado à rede;
- **Autenticação Mútua:** Quando os dispositivos se conectam somente quando ambos garantiram que os seus certificados são válidos;
- **Autenticação de Três Vias:** Quando há um administrador que irá associar os dispositivos e suas certificações, para garantir uma conexão segura entre os mesmos.

Vale lembrar que todos os formatos de autenticação apresentados são úteis, independente disso os mesmos precisam ser analisados e implementados em cada sistema conforme as suas necessidades específicas, ou seja, considerando as tecnologias, os dispositivos utilizados e os objetivos dos serviços operacionalizados.

Saiba Mais

Segurança é um tema muito sério e complexo, e que certamente mereceria um curso inteiro sobre o tema. No entanto, como isso não é possível, indicamos fortemente que você assista aos dois vídeos indicados. Os quais trazem as visões de experientes profissionais da área, que tratam de pontos importantíssimos envolvendo o tema.

- IoT Privacidade e Segurança
<https://www.youtube.com/watch?v=zp0NOXv6THY>
- Desafios de segurança em sistemas de Internet das Coisas (IoT) | Victor Hayashi
<https://www.youtube.com/watch?v=AybnhuXK9ME>

Nota:

O autor do segundo vídeo é também coautor de um livro recente e bastante claro sobre o tema - MORAES, Alexandre de; Hayashi, Victor Takashi. **Segurança em IoT: Entendendo os riscos e ameaças em IoT**. Alta Books, 2021. ISBN: 9788550816517

Inteligência Artificial e Aprendizado de Máquina em IoT

A Inteligência Artificial (IA) é um campo da ciência da computação que se concentra no desenvolvimento de sistemas e algoritmos capazes de executar tarefas que normalmente dependem da ação e inteligência humana. Essas tarefas incluem, entre outras:

- Raciocínio;
- Aprendizado;
- Resolução de problemas;
- Compreensão de linguagens;
- Percepção visual e
- Tomada de decisões.

Entre os objetivos da IA, está criar máquinas e programas de computador que possam se assemelhar ao pensamento humano, permitindo a execução de tarefas complexas de forma autônoma e eficiente. Isso é possível pois, por meio da manipulação de grandes quantidades de dados, que ao serem processadas por algoritmos especializados, identificam padrões e tomam decisões lógicas com suas inferências.

A IA e suas subáreas estão transformando a forma como trabalhamos, vivemos e interagimos com a tecnologia, e esta tem gerado melhorias significativas na automação das mais diversas atividades e na capacidade de tomada de decisões cada vez mais assertivas.

Curiosidades

A aplicação mais famosa e referência atualmente da IA é o *ChatGPT*, o qual como o próprio nome indica, trata-se de um *Chat* referindo-se à sua funcionalidade de interação textual *Chatbot* e *GPT* de *Generative Pre-trained Transformer* ou Gerador Transformador Pré-treinado.

Seu funcionamento se dá por meio de redes neurais artificiais, que são programas de computador projetados para processar e gerar texto natural. A operação da ferramenta é baseada em “aprendizado de máquina” e “processamento de linguagem natural”, o que proporciona respostas assertivas e complexas devido à precisão lógica e inteligente de seus algoritmos.

Embora as IAs já tenham alcançado avanços notáveis, elas ainda enfrentam desafios complexos, como compreensão de contexto, ética, regulação, interpretação, viés, preconceito, entre outros.

Para saber em detalhes sobre o funcionamento do *ChatGPT* e utilizá-lo na prática, acesse o site <<https://chat.openai.com/>> da empresa de pesquisa e implantação de IA - *OpenAI* que o desenvolveu.

Os avanços da IA dependeram de evoluções computacionais “hardware” e também da eficácia de muitos de seus métodos “software”, dentre elas há um especial destaque do subcampo de *Machine Learning* (ML) ou Aprendizado de Máquina e de algumas de suas subcategorias, tais como: *Natural Language Processing* ou Processamento de Linguagem Natural, *Deep Learning* ou Aprendizado Profundo, *Federated Learning* ou Aprendizado Federado e o *Reinforcement Learning* ou Aprendizado por Reforço.

Alguns exemplos de como a IA e o ML podem colaborar fortemente com a IoT, são elencados a seguir:

- **Análise de Dados:** Os sistemas IoT geram enormes volumes de dados provenientes dos seus dispositivos interconectados. Com a IA os dados podem ser analisados e processados em tempo real, detectando possíveis anomalias e assim decidir-se ações para cada ocasião.
- **Otimização de Processos:** Pode-se otimizar processos em ambientes IoT, ajustando-se automaticamente parâmetros do sistema com base no histórico das condições dos mesmos e nos requisitos solicitados nos processos, gerando economia de recursos e reduzindo custos.
- **Processamento de Imagens:** Pode ser empregada para analisar imagens capturadas por câmeras, podendo incluir diversas funções como o reconhecimento facial, detecção de gestos, detecção de objetos e leitura de placas de veículos.
- **Segurança:** Podem interagir com sistemas de criptografia com ações personalizadas pelo usuário, identificando ameaças de segurança, ajudando a inibir atividades suspeitas e gerando respostas a tais ocasiões de forma mais eficaz.

Graças em parte aos avanços dos métodos de IA, dispositivos e sensores em geral têm conseguido trabalhar de forma mais “inteligente”, com melhor eficiência e com riscos à segurança bastante minimizados.

Aprendizado de Máquina

É o estudo de padrões para o desenvolvimento de IA, os quais possibilitam sistemas lógicos a aprender com a base em dados coletados. Exemplos desse tipo de aprendizado são os sistemas já existentes em dispositivos com reconhecimento facial, reconhecimento de fala, identificação de imagens entre outros.

Através da codificação de programas em linguagens como *Python*, é possível chegar a diferentes cenários de experiências as quais podem contribuir para o desenvolvimento de aplicações capazes de tomar decisões mais próximas das humanas.

Um exemplo de aplicação integrando IA e ML em IoT, são os carros autônomos. Os quais através dos seus dispositivos (sensores/atuadores), podem determinar qual é a velocidade permitida da pista em que o percurso ocorre, seguir a rota determinada pelo GPS realizar ligações através de reconhecimento de voz por *bluetooth* e muitas outras funções que estiverem disponíveis. Ao longo do tempo é possível também que os algoritmos implementados nos programas, aprendam com os padrões de comportamento do motorista e com esse conhecimento adquirido, sendo então utilizado pelo próprio veículo para proporcionar experiências personalizadas.

Como citado anteriormente, existem vários métodos para alcançar IAs altamente eficientes, sendo o ML um dos mais evidentes e utilizados dada a sua eficácia. O Aprendizado de Máquina envolve treinar modelos de IA, com conjuntos de dados para que eles possam aprender com exemplos e melhorar seu desempenho ao longo do tempo, ou seja, aprendizado de máquina é o estudo de padrões para desenvolvimento máquinas “hardware/software” inteligentes, o qual possibilita ao sistema lógico aprender com a base nos conjuntos de dados coletados.

Saiba Mais

A seguir sugerimos que você assista a três vídeos disponíveis no *YouTube*, os quais trazem exemplos de aplicações que utilizam IA e ML em IoT e na indústria.

- **Tecnologias 4.0 - Sistema MES, IoT e Inteligência Artificial**
<https://www.youtube.com/watch?v=p3Dt7zFNkjh>
- **Você já está aplicando Machine Learning (ML) em seus projetos de IoT?**
<https://www.youtube.com/watch?v=H1HaEK31L54>
- **Machine Learning in IoT solutions**
<https://www.youtube.com/watch?v=R0S9DfzxY&t=3s>

Edge Computing

Edge Computing significa computação de borda, isso porque o objetivo dessa infraestrutura distribuída é aproximar os dados do objetivo final no processo, ou seja, das fontes destes dados. No caso de processos de infraestrutura IoT, o deslocamento dos dados dos dispositivos e sensores até a base de dados, armazenando de forma segura e rápida as informações coletadas.

Grandes quantidades de dados podem ser geradas por alguns dispositivos IoT em infraestrutura *Edge*, sendo a baixa latência essencial para muitos desses casos de uso, com o processamento de dados em borda sendo fundamental para o sucesso da administração do sistema.

Os benefícios da implementação de uma infraestrutura *Edge*, são a tomada decisões rápidas e assertivas, a flexibilidade de implementação e o *Backhaul* que é uma parte importante da infraestrutura de rede, pois conecta o dispositivo final “coisa”, à central de rede garantindo uma transferência eficiente de dados em diferentes configurações topológicas.

Por estar em contato direto com os dispositivos que estão na borda (*edge devices*), a aplicação do *Edge Computing* torna-se essencial para economizar no desenvolvimento da infraestrutura, e assim portanto reduzindo o *Backhaul*, o qual é uma parte crítica da arquitetura pois garante que os dados coletados dos *edge devices* sejam transferidos de forma eficiente e confiável para onde precisam ser processados.

O *Backhaul* pode envolver diferentes tecnologias de rede, como cabos de fibra ótica, conexões sem fio de alta velocidade ou conexões tradicionais, dependendo das necessidades específicas do ambiente e das aplicações de *Edge Computing*. Sendo essencial por ser a “via de retorno” pela qual os dados coletados pelos *edge devices* são transmitidos de volta para processamento, armazenamento ou análise centralizada.

Fog Computing

Para que o processamento de dados seja eficaz com a utilização da computação de borda, é necessário a utilização da Computação em Névoa ou *Fog Computing*, a qual possibilita distribuir todos os recursos disponíveis em nuvem (*Cloud Computing*) até a borda do sistema (*Edge Computing*), tornando possível o acesso ao armazenamento, controle de rede e serviços de computação para todo o sistema IoT, o que resulta no aumento do poder computacional entre a nuvem e os dispositivos.

Além disso, também visa tornar os armazenamentos, processamentos e análises de informações mais veloz e ágil, melhorando o tempo de resposta, reduzindo a quantidade de dados enviados para a nuvem e diminuindo a latência da rede e da Internet, ou seja, a ideia é a de descentralização, com o processamento dos dados ocorrendo mais proximamente de onde os mesmos são capturados.

Para sistemas avançados de IoT é necessário que toda a infraestrutura aplicada esteja em harmonia entre os serviços de Nuvem (*Cloud Computing*), Névoa (*Fog Computing*) e Borda (*Edge Computing*), pois todos os dados coletados na borda pelos dispositivos e sensores podem ser enviados para nuvem através da névoa, e essas ações que só poderiam ser realizadas em nuvem, agora estão fácil acesso na borda através da névoa. Isso gera confiabilidade nos dados coletados e agilidade para procedimentos emergenciais manutenções corretivas nos dispositivos e sensores, e propicia ainda ações de proteção contra ataques ao sistema conectado.

A figura a seguir ilustra a como computação em névoa atua como intermediária entre a nuvem e os dispositivos finais, aproximando serviços de processamento, armazenamento e rede dos próprios dispositivos finais, os quais recebem a denominação de nós da névoa.

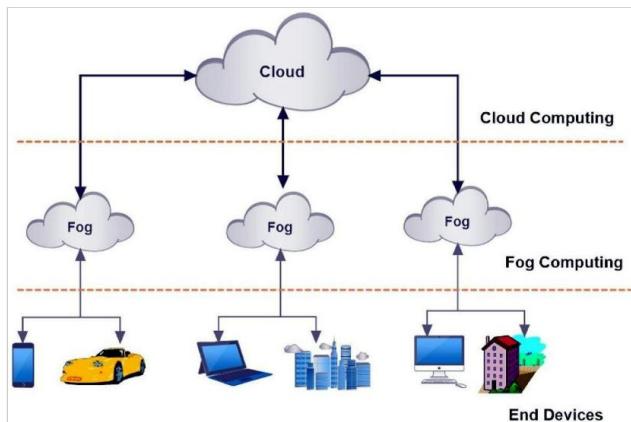


Figura 27. A *Fog Computing* é uma extensão da *Cloud Computing*, mas está mais próxima dos dispositivos finais (ATLAM; WALTERS; WILLS, 2018).

Saiba Mais

Esperamos que esteja gostando dos temas trazidos em nosso curso, estamos esperançosos de que tudo tenha ficado o mais claro possível. Porém temos consciência de que tratamos de temas que por vezes são bastante complexos, os quais por mais que nos esforcemos, podem deixá-lo(a) com alguma dúvida.

Visando então complementar o que você aprendeu, sugerimos que assista aos vídeos que indicamos a seguir, os quais apesar de curtos, temos certeza, lhe ajudarão a fixar importantes conceitos, bem como lhe trarão boas ideias sobre como implementar soluções IoT fazendo uso de *Edge* e *Fog Computing*.

- *Edge Computing*: o processamento mais rápido de dados
<https://www.youtube.com/watch?v=EqFU2d44d50&t=15s>
- *What is edge computing, and why is it so important?* - InHandgo.com
https://www.youtube.com/watch?v=R9P0C_E1yxQ
- *Fog Computing na IoT*
<https://www.youtube.com/watch?v=VRoyQbeFOb4>
- *Conceito de planta industrial Fog computing, Cloud computing*
<https://www.youtube.com/watch?v=kNsTYPjsdZ0>

Aplicações de *Blockchain* em IoT

Dentro dos conceitos abordados quanto à evolução em sistemas IoT, a associação com *Blockchain*¹ representa bem o de “avançado tecnológico” e industrial. Sendo que segundo especialistas da área, em um futuro próximo, sistemas IoT em conjunto com *Blockchain* serão tão comuns e necessários quanto as condições tecnológicas atuais.

O que é *Blockchain* e como podemos aplicá-lo em IoT

O surgimento da primeira [criptomoeda](#)² em 2008, o *Bitcoin*, trouxe consigo a possibilidade de se ter uma moeda confiável e descentralizada. O princípio de funcionamento desse sistema de proteção e transação de dados, se dá por blocos de dados, por isso nome *Blockchain* (corrente de blocos).

Trata-se de uma tecnologia que agrupa um conjunto de informações que se conectam por meio de criptografia. Assim, transações financeiras e outras operações podem ser feitas de forma segura, tornando todo o processo mais robusto contra fraudes e clonagens digitais, pois, cada uma delas é registrada e rastreada desde a sua criação.

Em paralelo, também há o benefício da transparência e confiabilidade que o sistema oferece, sendo o conceito de descentralização dos dados coletados responsável por proporcionar maior autonomia ao usuário do sistema. Administrar os dados produzidos pela IoT e as implicações relacionadas aos mesmos pode ser um processo complexo e surpreendentemente diversificado devido às suas possíveis variedades de opções, por isso a combinação entre sistemas IoT e *Blockchain* são tão necessários.

O fluxo de dados de uma empresa que utiliza sensores e dispositivos IoT é imenso e o *Blockchain* ajudará a resolver questões de transparência, conferindo o *status* de inalterabilidade dos dados e informações transacionadas e armazenadas nas redes utilizadas.

Em todas elas o *Blockchain* colaborará para aumentar a segurança dos dados transacionados e armazenados, tornando as informações imutáveis devido à sua característica de coletas descentralizada, culminando assim em chances de ocorrência de invasões, vazamentos ou alterações de informações quase inexistentes e isso com o benefício extra de que todo o conjunto de transações pode ser facilmente verificado e/ou auditado.

Você Sabia

A seguir indicamos, alguns exemplos práticos em vídeo disponíveis no *YouTube*, os quais irão propiciar uma melhor compreensão frente ao *Blockchain* em IoT.

- **Blockchain e IoT: Identificação digital de ativos reais**
<https://www.youtube.com/watch?v=rBbaogxIVeE>
- **Tecnologia blockchain e suas aplicações**
https://www.youtube.com/watch?v=37ufm8_waj4
- **IOT e Blockchain**
<https://www.youtube.com/watch?v=5NAKqnleN9A>

¹ O *Blockchain* (que não possuía este nome na época) surgiu em 1991 nos laboratórios da Xerox Parc, quando os cientistas da computação Stuart Haber e W. Scott Stornetta idealizaram uma solução computacional que permitia o registro de datas em documentos digitais, de modo que essa informação não pudesse ser modificada de forma alguma. O trabalho dos autores foi divulgado através do artigo "How to Time-Stamp a Digital Document".

² Criptomoeda ou cibermoeda é um meio de troca, geralmente descentralizada, que se utiliza das tecnologias de *blockchain* e criptografia para assegurar a validade das transações e a criação de novas unidades da moeda.

Ainda sobre *Blockchain*

Blockchain é como um livro de registros, só que digital e muito seguro, porém, compartilhado via rede, onde várias pessoas podem anotar informações, mas ninguém pode apagar ou alterar o que foi escrito. Essas informações são organizadas em blocos que são vinculados uns aos outros, criando uma cadeia (*chain*) rastreada de dados. Entre as principais características do *Blockchain*, elenca-se as que se seguem:

- **Descentralização:** Não é controlado por uma única entidade, mas por uma rede de computadores (nós) que validam e mantêm informações;
- **Imutabilidade:** Uma vez que as informações são registradas em um bloco, elas não podem ser alteradas sem o consenso da rede. Em relação à segurança, por utilizar métodos de criptografia avançados e com chaves criptográficas públicas e privadas variando entre 160 e 256 bits, é difícil para qualquer pessoa falsificar ou corromper os dados; e
- **Transparência:** Os registros são visíveis para todos os participantes da rede, garantindo portanto a transparência dos mesmos.

Alguns motivos pelos quais o *Blockchain* é tão importante para aplicações atuais e futuras em IoT

- **Identificação e Autenticação Segura:** O *Blockchain* cria identidades seguras para dispositivos IoT, com cada dispositivo possuir uma identidade única registrada imutável, o que evita a falsificação e torna o sistema de autenticação seguro, pois garante que apenas dispositivos autorizados tenham acesso à rede.

- **Registro de Dados:** Os sensores IoT podem coletar dados e registrar automaticamente na *Blockchain*. Isso cria um histórico seguro e confiável das informações.
- **Rastreabilidade:** O *Blockchain* permite rastrear a origem e o histórico de produtos, isso é útil em casos como grandes centros logísticos, onde você pode verificar a autenticidade e a jornada de um produto.
- **Pagamentos e Micropagamentos:** O *Blockchain* facilita transações seguras e eficientes entre dispositivos IoT. Isso é útil em sistemas de pagamento entre máquinas ou para cobrar pelo uso de serviços (área de origem e maior aplicação).
- **Segurança:** A segurança é fundamental na IoT, o *Blockchain* pode proteger contra ataques cibernéticos, garantindo que os dados não sejam adulterados.
- **Compartilhamento de Dados:** Permite que diferentes partes compartilhem dados de IoT com confiança e sem a necessidade de intermediários.

Em resumo, *Blockchain* é uma tecnologia que oferece segurança, transparência e confiança aos dados da IoT, tornando-a uma solução valiosa para diversas aplicações. Ela cria um ambiente em que os dispositivos podem interagir e compartilhar informações de forma segura e automatizada, tornando a IoT mais eficiente e confiável.

É importante destacar que a implementação bem-sucedida de *Blockchain* em IoT requer uma consideração cuidadosa dos requisitos de segurança, escalabilidade e eficiência, bem como a escolha da plataforma *Blockchain* apropriada. Além disso, a conformidade regulatória é um fator crítico a ser considerado, especialmente em setores altamente regulamentados como a saúde, a indústria aeronáutica e o mercado financeiro por exemplo.

Breve despedida

À medida que concluímos nossa jornada pelo amplo campo da Internet das Coisas com foco em *Python*, ficou evidente que estamos diante de um horizonte repleto de oportunidades e inovações. *Python*, com sua sintaxe clara e bibliotecas robustas, se destaca como uma escolha ideal para desenvolvedores e entusiastas da área.

As possibilidades em IoT são uma força transformadora que está moldando nossa sociedade e economia. E como visto durante o curso, a linguagem *Python* integrada a ferramentas como *Arduino* e *Raspberry Pi*, são excelentes opções de peças-chave para desbloquear esse potencial.

Compreendemos ainda os benefícios oferecidos pela *Cloud Computing* e a importância da escolha criteriosa entre as várias opções de plataformas IaaS, PaaS e SaaS, bem como os benefícios e as facilidades operacionalizadas através da utilização de APIs.

Trouxemos à tona a sempre premente questão da segurança presente em qualquer sistema, conjuntamente com os importantes temas sobre criptografia e autenticação. Tratamos dos benefícios trazidos pela IA e o ML, e de como a *Edge Computing* e a *Fog Computing* podem contribuir para redução de custos e o aumento da segurança ao estruturarem de forma otimizada servidores e dispositivos, por fim trouxemos ainda algumas das benéficas questões relacionadas à associação entre os sistemas IoT e *Blockchain*.

Ao chegarmos até aqui, esperamos que este curso tenha despertado em você mais do que apenas uma inspiração inicial, e, portanto, desejamos que continue aprendendo, experimentando e criando soluções IoT inovadoras que ajudem de fato, a moldar o futuro da conectividade auxiliando nossa sociedade a se tornar mais conectada, justa e "inteligente"!

Tags do conteúdo

IoT, Internet das Coisas, Conectividade, Dispositivos Inteligentes

Sensores, C++, Python, Tinkercad, Arduino, Raspberry Pi

Bibliotecas Python, Arquitetura em IoT, Cloud Computing

Fog Computing, Edge Computing, Gateway, Sensores IoT

Serviços em Nuvem, API, Latência, Back-end

Bluetooth, NFC, 4G, 5G, Wi-Fi

PaaS, IaaS, SaaS, Conectividade, Armazenamento de Dados

Dispositivos IoT, XML, YAML, JSON, REST

Serialização de Dados, HTTP, WEB, CRUD

Códigos Maliciosos, Vírus, Criptoanálise, Criptografia

Autenticação de Identidade, Autenticação de Origem de Dados, Transmissão Segura

AES, TLS/SSL, Senhas, PIN, Token, Biometria

Autenticação Unilateral, Autenticação Mútua, Autenticação de Três Vias

Inteligência Artificial, IA, Machine Learning, Identificação de Padrões

Referências

ATLAM, Hany F.; WALTERS, Robert J.; WILLS Garry B. **Fog Computing and the Internet of Things: A Review.** *Big Data and Cog Computing* 2,no. 2: 10. DOI: <<https://doi.org/10.3390/bdcc2020010>>

CÉZAR, Sillas de Oliveira. **Uso de Python e conceitos de internet das coisas aplicados a técnicas vibro-acústicas clássicas p detecção de vazamentos.** Dissertação (Mestrado em Engenharia Mecânica). Faculdade de Engenharia de Bauru, Universidade Es Paulista, Bauru, p. 104, 2021.

CURVELO, Rakky. **Conheça 5 aplicações de blockchain nas empresas.** 09 nov. 2022. Disponível em: <<https://br.hubspot.com/blog/marketing/aplicacoes-blockchain>>. Acesso em: 30 de Ago. 2023.

FACHINI, Moisés Panegassi et al. **Internet das coisas: Uma breve revisão bibliográfica.** Revista Conexões - Ciência e Tecnologia Fortaleza, v.11, n. 6, p. 85-90, 2017.

FILHO, Mauro Faccioni. **Internet das Coisas.** Livro digital. Palhoça: UnisulVirtual, 2016.

KHAN, Minhaj Ahmad; SALAH, Khaled. **IoT security: Review, blockchain solutions, and open challenges.** *ScienceDirect - Future Generation Computer Systems* v.82, p 395 - 411, 2018. Disponível em: <<https://www.sciencedirect.com/science/article/abs/pii/S0167739X17315765?via%3Dihub>>. Acesso em: 24 de Ago. 2023.

MENDONÇA, Cláudio Márcio Campos; ANDRARE, António Manuel Valente de; NETO, Manoel Veras de Sousa. **Uso da IoT, big dat inteligência artificial nas capacidades dinâmicas.** RPCA - Revista Pensamento Contemporâneo em Administração. Rio de Janeir v.12, n. 1, p. 131-151, 2018.

NAKAMOTO, Satoshi. **Bitcoin: A Peer-to-Peer Electronic Cash System.** Bitcoin Glossary: 2018 Annual National Seminar. Disponív <https://www.ussc.gov/sites/default/files/pdf/training/annual-national-training-seminar/2018/Emerging_Tech_Bitcoin_Crypto.pdf>. Ace em: 29 de Ago. 2023.

OLIVEIRA, Sérgio de. **Internet das Coisas: Com ESP8266, Arduino e Raspberry PI.** 1ª Edição. São Paulo: Novatec Editora Ltda.

ROCHA, Isabela Fernandes da; KASSIMOTO, Kumiku Oshio. **Barreiras e benefícios na adoção de inteligência artificial e IoT na gestão da operação.** RAM - Revista de Administração Mackenzie. São Paulo, n.23 (4), 2022.

SILVA, Thalita Bento da; PESSOA, Claudio Roberto Magalhães; JAMIL, George Leal; JUNIOR, Mário Márcio Figueiredo Rosa. **Inter Things: Concepts, Applications, Challenges and Trends.** In: 13th CONTECSI - International Conference on Information Systems Technology Management, 2016, São Paulo.

SINGER, Talyta. **Tudo conectado: conceitos e representações da internet das coisas.** In: SIMSOCIAL - Simpósio em Tecnologia Digitais e Sociabilidade n.02, 2012, Salvador.

VIEIRA, Moroni Neves. **Medições e avaliações comparativas de desempenho e energia de algoritmos de machine learning pa mitigar ameaças de disponibilidade em ambiente IoT.** Dissertação (Mestrado em Tecnologia da Informação). Instituto Federal de Educação, Ciência e Tecnologia da Paraíba - IFPB, João Pessoa, p.125, 2021.

Referências das Figuras

Figuras de 1 a 7 e 9: *Freepik* <<https://br.freepik.com/vetores-gratis/>> e <<https://img.freepik.com/fotos-gratis/>>. Acesso entre: Ago. e 2023.

Figura 8: Ardu Robótica <https://www.ardurobotica.com.br/MLB-1353914165-mq-7-modulo-sensor-gas-mq7-monoxido-carbono-e-arduino_-JM>. Acesso em: 04 de Set. 2023.

Figuras 10 e 11: Autodesk <<https://www.tinkercad.com/>>. Acesso em: 11 de Set. 2023.

Figuras 12, 13, 14, 15, 16, 17, 18, 19 e 20: Fabrica de Conteúdos <<https://fabricadeconteudos.com.br/>>. Criadas entre Ago. e Set .

Figura 21:Microsoft <<https://azure.microsoft.com/pt-br/>>. Acesso em: 09 de Set. 2023.

Figura 22:IBM <<https://www.ibm.com/cloud>>. Acesso em: 26 de Ago. 2023.

Figura 23: Amazon Web Services <<https://aws.amazon.com/pt/>>. Acesso em: 19 de Ago. 2023.

Figura 24:Heroku <<https://www.heroku.com/>>. Acesso em: 16 de Ago. 2023.

Figura 25:Oracle <<https://www.oracle.com/br/cloud/>>. Acesso em: 22 de Set. 2023.

Figura 26:Google <<https://cloud.google.com/appengine>>. Acesso em: 22 de Set. 2023.

Nota

Todas as marcas citadas e/ou exibidas neste material, pertencem aos seus respectivos fabricantes e/ou desenvolvedores

Design by Fábrica de Conteúdos Educação ©