CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Universidade Federal
de Campina Grande

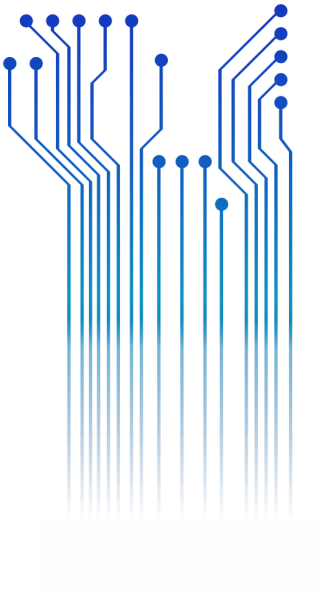
Centro de Engenharia
Elétrica e Informática

JOÃO CARLOS SOUZA PINTO

Departamento de
Engenharia Elétrica

TRABALHO DE CONCLUSÃO DE CURSO

INDUSTRIAL INTERNET OF THINGS: SURVEY E APLICAÇÃO PARA CONTROLE DE ACESSO



Campina Grande
2017

João Carlos Souza Pinto

INDUSTRIAL INTERNET OF THINGS: SURVEY E APLICAÇÃO PARA CONTROLE DE ACESSO

Trabalho de Conclusão de Curso submetido à Unidade Acadêmica de Engenharia Elétrica da Universidade Federal de Campina Grande como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciências no Domínio da Engenharia Elétrica.

Área de Concentração: Internet das Coisas Industrial

Orientador:
Professor Rafael Bezerra Correia Lima, D.Sc

Campina Grande
2017

João Carlos Souza Pinto

INDUSTRIAL INTERNET OF THINGS: SURVEY E APLICAÇÃO PARA CONTROLE DE ACESSO

Trabalho de Conclusão de Curso submetido à Unidade Acadêmica de Engenharia Elétrica da Universidade Federal de Campina Grande como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciências no Domínio da Engenharia Elétrica.

Área de Concentração: Controle e Automação

Aprovado em / /

Professor Avaliador

Universidade Federal de Campina Grande

Avaliador

Professor Rafael Bezerra Correia Lima, D. Sc.

Universidade Federal de Campina Grande

Orientador, UFCG

Dedico este trabalho aos meus pais e minhas irmãs, que sempre me apoiaram e ficaram do meu lado, diante de todas as dificuldades do curso.

AGRADECIMENTOS

Agradeço aos meus pais, Olavo e Célia, em primeiro lugar, por todo o esforço e trabalho que tiveram para que eu pudesse ter uma boa educação e saúde. Meus queridos pais, que nunca deixaram que faltassem oportunidades na minha vida para que eu pudesse crescer, foram essenciais na minha formação.

Agradeço também às minhas irmãs Marianna, Priscila e Camila por sempre se preocuparem tanto comigo, me ligando sempre que algum problema surgia ou simplesmente por conta da saudade.

Agradeço a Ana Loísa, minha namorada, por estar sempre presente mesmo estando tão longe, e por ter me acompanhado em praticamente toda a minha jornada neste curso.

Agradeço a todos os meus amigos, que estiveram comigo durante todos esses anos e que foram indispensáveis para que eu aguentasse passar tanto tempo longe da minha família e namorada.

Agradeço também ao professor Rafael Bezerra Correia Lima por ter aceitado a me orientar neste trabalho.

Por fim, agradeço a todos que, de alguma forma, passaram pela minha vida e contribuíram para a minha formação pessoal e acadêmica.

Resumo

Estamos vivendo em uma época onde os mais diferentes objetos estão se conectando com a internet, como geladeiras, televisores, torradeiras, automóveis e várias outras coisas. Internet das Coisas, como é chamada, trata-se de uma rede de objetos físicos que possuem tecnologia embarcada necessária para coletar e transmitir dados através da internet. Trazendo essa tecnologia para o cenário das industrias, a mesma recebe o nome de Internet das Coisas Industrial, uma rede que promete conectar bilhões de dispositivos com a internet com a finalidade de facilitar o modo como as indústrias funcionam e trazer benefícios para a população.

Este trabalho de conclusão de curso tem o propósito de, em um primeiro momento, expor o estado da arte da Internet das Coisas no cenário da indústria para que o leitor possa entender a importância dessa tecnologia, principalmente na área de automação. Em um segundo momento, será apresentado também uma aplicação desenvolvida pelo autor com o objetivo de demonstrar uma das possibilidades que pode ser implementada na indústria graças à Internet das Coisas.

Palavras-chave: Internet das Coisas, Internet das Coisas Industrial, MQTT, NFC, Automação industrial, Automação predial.

Abstract

We are living in an era in which different objects are connecting to the internet, like refrigerators, television, toasters, automobiles and a lot of other things. The Internet of Things is a network of physical objects that have enough built-in technology to collect and send data through the internet. When we bring this technology to the industrial scenario, it's given the name of Industrial Internet of Things, a network that promises to connect billions of devices through the internet with the aim of facilitating the way industry works and bring benefits to the population.

The first part of this final paper intends to show the state of art of the Industrial Internet of Things, so that the reader can understand that this technology is really important, mainly in the automation area. The second part of this final paper will show an application developed by the author to show one of several possibilities that the Industrial Internet of Things has to offer.

Keywords: Internet of Things, Industrial Internet of Things, MQTT, NFC, Industrial automation, Building automation.

Lista de Ilustrações

Lista de tabelas

Lista de Abreviaturas e Siglas

Internet das coisas	IoT
Industrial internet of things	IIoT
Laboratorio de Instrumentacao Eletronica e Controle	LIEC
Near Field Communication	NFC
Identification	ID
Message Queuing Telemetry Transport	MQTT
Machine to Machine	M2M
Radio-Frequency Identification	RFID
Internet Protocol	IP
Internet Protocol version 4	IPv4
Internet Protocol version 6	IPv6
EXtensible Messaging and Presence Protocol	XMPP
Data Distribution Service	DDS
Application Programming Interface	API
Sercice Oriented Architecture Protocol	SOAP
Representational State Transfer	REST

Sumário

1. INTRODUÇÃO

Em um futuro não muito distante, presume-se que os dispositivos eletrônicos estarão conectados com a internet, o que tornará possível a comunicação entre os dispositivos para diversas finalidades como, por exemplo, a troca de informações entre si. A Internet das Coisas (IoT) tornou-se um tema bastante recorrente nos últimos anos e atrai atenção da comunidade acadêmica por ser um tema relativamente novo e possuir inúmeras aplicações. A *Industrial Internet of Things* (IIoT) é a parte da internet das coisas focada em dispositivos e objetos utilizados em ambientes industriais. A IIoT possibilita uma melhor visibilidade e compreensão das operações de uma empresa através da integração de sensores, middleware, software e computação na nuvem.

Este trabalho tem o propósito de dissertar sobre o estado da arte da IIoT, abordando tópicos importantes que influenciam na sua implementação, a necessidade do uso de tecnologias já existentes, a escolha do protocolo de comunicação e as barreiras encontradas até o momento. A tecnologia IoT começa aos poucos a ser estudada e desenvolvida na UFCG, no Laboratório de Instrumentação Eletrônica e Controle (LIEC), enquadrados pelos professores Péricles Barros, George Acioli e Rafael Bezerra

A fim de ilustrar as potencialidades da internet das coisas, foi desenvolvido um sistema distribuído de controle de acesso, capaz de gerenciar o fluxo de pessoas nas salas do LIEC.

O sistema permite a entrada de usuários credenciados, mediante um cadastro na nuvem. O acesso é liberado por meio de um leitor de tags NFC (Near Field Communication).

1.1. Objetivos

O principal objetivo deste trabalho é dissertar sobre a *Industrial Internet of Things* (IIoT), bem como integrar os controles de acesso à internet, com a

finalidade de demonstrar uma aplicabilidade do tema abordado em um cenário atual e presente no dia a dia de muitas pessoas.

1.2. Estrutura do trabalho

A primeira etapa do projeto consiste em uma pesquisa bibliográfica mais a fundo a respeito da *Industrial Internet of Things*, com o intuito de ilustrar seu estado da arte. Nessa pesquisa, serão abordados temas como as tecnologias e protocolos mais aptos para a comunicação de dispositivos entre si, bem como os consórcios atuais, os padrões já consolidados, os fabricantes que já se encontram no mercado, as barreiras a serem enfrentadas, etc.

Em uma segunda etapa, foi projetado e desenvolvido um sistema de controle de acesso distribuído. Foram utilizadas as seguintes plataformas: Arduino Uno, juntamente com um leitor NFC e, para conectá-los à internet, um NodeMCU. A comunicação dos dispositivos com o servidor foi feita através do protocolo *publish/subscribe* Message Queuing Telemetry Transport (MQTT), que por ser leve e de simples implementação é bastante empregado no ramo. O *broker* utilizado para a comunicação foi embarcado em um Intel NUC.

Em uma terceira etapa, o módulo de acesso foi testado e mostrou-se funcionando como desejado, restando agora apenas integrá-lo à porta para que seja possível fazer uso da ferramenta criada.

1.3. Visão Geral do Texto

Neste capítulo do trabalho estão descritas, de forma resumida, algumas informações sobre o estado da arte da *Industrial Internet of Things (IIoT)*, o protocolo MQTT, o Arduino Uno, o leitor NFC e o NodeMCU, que são úteis para o entendimento do projeto.

2. INDUSTRIAL INTERNET OF THINGS (IIOT)

A internet surgiu no auge da Guerra Fria, na década de 60, com o objetivo de criar um sistema de compartilhamento de informações para aprimorar as estratégias de guerra; hoje, no entanto, a internet entra numa nova etapa da sua existência, a chamada Internet das Coisas (IoT). [1]

O termo “coisas” de Internet das Coisas se refere aos objetos presentes no dia a dia de todos que estão interconectados com a internet, tais como eletrodomésticos, sensores, ar-condicionado, meios de transporte, industrias, etc. Existe uma parte da IoT que é focada em objetos e dispositivos utilizados no meio industrial, essa parte é denominada *Industrial Internet of Things,* ou simplesmente IIoT. [2]

O boom da IIoT é bastante recente, isso se dá graças ao grande avanço das tecnologias necessárias para a implementação dessa tecnologia. Os protocolos de comunicação estão mais avançados, a tecnologia de armazenamento na nuvem está mais barata e vários outros fatores contribuíram para que a IIoT passasse a ser mais bem vista no setor industrial e pelas grandes empresas.

2.1. Casos de uso da IIoT

O potencial da Industrial Internet é enorme e possui vasta aplicabilidade em diferentes áreas de produtividade, tais como logística, aviação, transporte, saúde, produção de energia e muitos outros. Quando se trata da IIoT, não existe uma solução única que sirva para todas as aplicações, portanto identificar e traçar uma estratégia para cada oportunidade pode não ser tão simples [2]. Na figura 1, é ilustrado como a IIoT engloba tantos setores.

Figura 1: Setores que a IIoT engloba

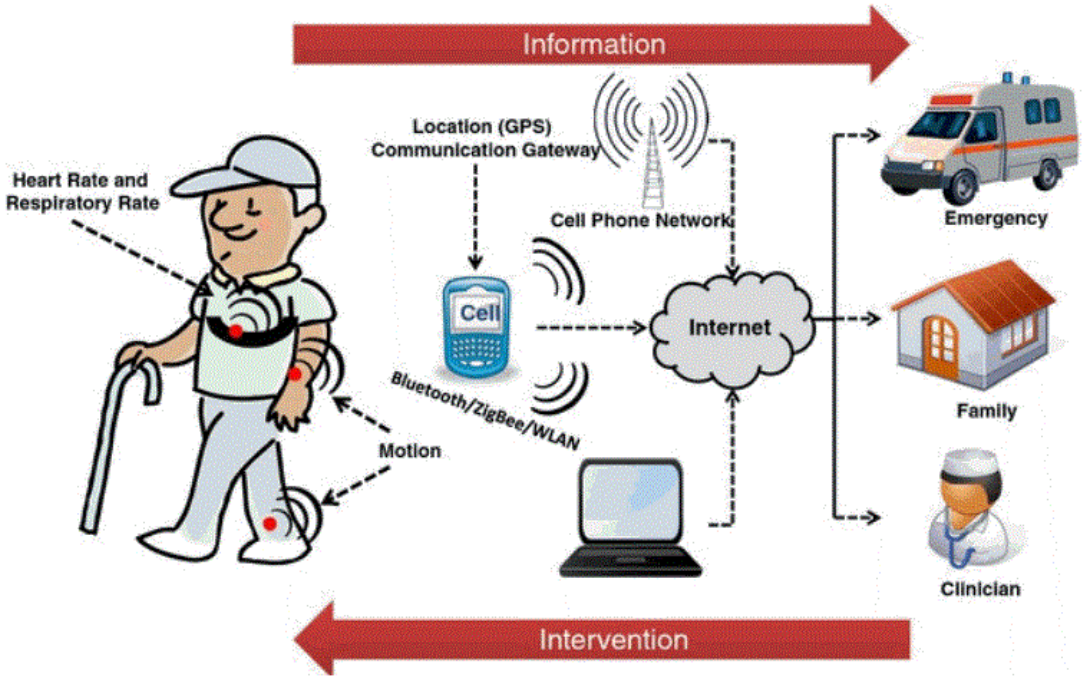
Fonte: http://industrialin.com/news/how-industrial-internet-things-changing-manufacturing-landscape

2.1.1. Saúde

A tecnologia IoT pode ser introduzida no campo da saúde em diversos setores, com várias possibilidades de aplicações (figura 2). Smartphones e sensores estarão interconectados através da internet para que seja possível aprimorar o cuidado com a saúde da população. Abaixo estão citadas algumas das possíveis aplicações IoT na área da saúde. [3]

- Sensor de nível de glicose;
- Monitoramento de eletrocardiogramas;
- Monitoramento da pressão sanguínea;
- Monitoramento da temperatura corporal;
- Monitoramento da saturação de oxigênio;
- Sistema de reabilitação;
- Gerenciamento de medicamentos;
- Gerenciamento de cadeiras de rodas;
- Soluções de cuidado com a saúde via smartphones.

Figura 2: Uso da IIoT para salvar mais vidas



Fonte: http://www.arch.ie/blog-post/iot-in-healthcare/

2.1.2. Industrias de Óleo e Gás

A coleta de dados é uma das principais razões pela qual a indústria de óleo e gás está investindo uma pequena fortuna na implementação de tecnologias IoT. A multinacional Shell investiu $87,000 em tecnologia para monitoramento digital em alguns de seus campos de petróleo na Nigéria; o investimento realizado no ano de 2015 possibilitou à Shell economizar um milhão de dólares, graças à redução de visitas nos campos de petróleo e expedição para coleta de dados. [4]

O uso de robôs e sensores na exploração, na previsão da produção de petróleo e gás, na manutenção de equipamentos e a otimização operacional são alguns dos outros motivos para o investimento na Internet das Coisas por parte dessas indústrias. [5]

2.1.3. Logística

Entre tantos casos de uso da IIoT na logística das empresas, um dos principais é o gerenciamento da cadeia de suprimento, em que as técnicas de analise preditiva da Big Data realmente aparece. As grandes companhias de logística do mundo precisam estar cientes dos últimos eventos, tais como política, condições climáticas locais que afetam as rotas de comércio, greves nos portos, etc. [2]

Através da análise da Big Data em tempo real e, portanto, a melhoria na previsibilidade dos eventos que influenciam a logística, será possível trocar as rotas em tempo real, substituir o meio de transporte das mercadorias para mitigar as ações das greves ou indesejáveis eventos climáticos que atrasariam a entrega dos produtos (figura 3).

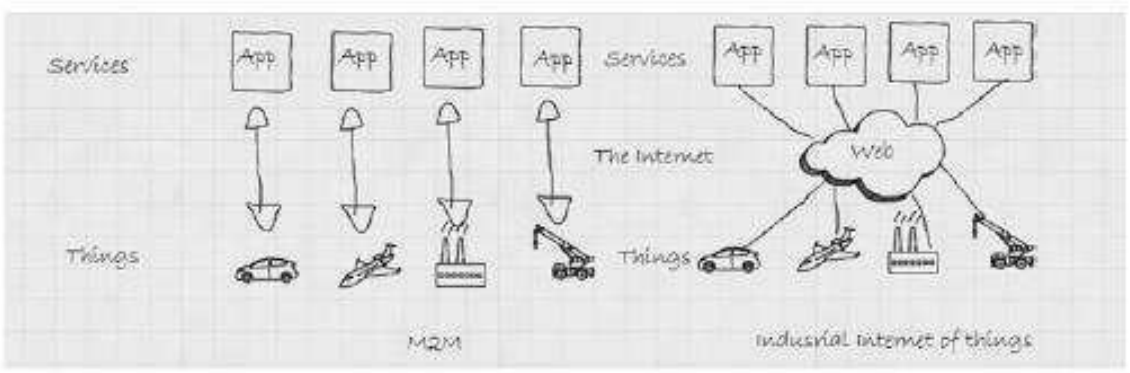
Figura 3: Influência da IIoT na logística das grandes companhias

Fonte: https://events.eft.com/iot/

2.2. IIoT vs M2M

Quando se fala de IIoT, é comum a confusão entre IIoT e Machine to Machine (M2M). A tecnologia Machine to Machine normalmente é associada a uma solução pontual, ou uma grande quantidade de um mesmo número de coisas, a exemplo de sistemas de sensores de temperatura, sensores de fluxo em uma refinaria, etc. Ao comparar superficialmente IIoT com Machine to Machine, é plausível imaginar que a Industrial Internet of Things é o arco que conecta os pilares da M2M, ou seja, a IIoT tenta juntar os sistemas separados em um grande e único sistema através da internet (figura 4). [6][7]

Figura 4: A IIoT ligando a M2M à web



Fonte:LIVRO

Na tabela abaixo, é possível identificar as principais diferenças entre o modelo M2M e a IIoT.

Tabela 1: Principais diferenças entre M2M e IIoT

M2M	**IIoT**
Comunicação ponto a ponto, geralmente embarcada no hardware, no local do cliente	Dispositivos comunicam-se entre si usando redes IP, incorporando uma variedade de protocolos de comunicação
Vários dispositivos usam rede com fio ou celular	Os dados são entregues através de uma camada de transmissão hospedada na nuvem
Os dispositivos não necessariamente precisam estar conectados à internet.	Muitos dispositivos requerem uma conexão ativa, em tempo real, com a internet.
Opções limitadas de integração	Opções ilimitadas de integração

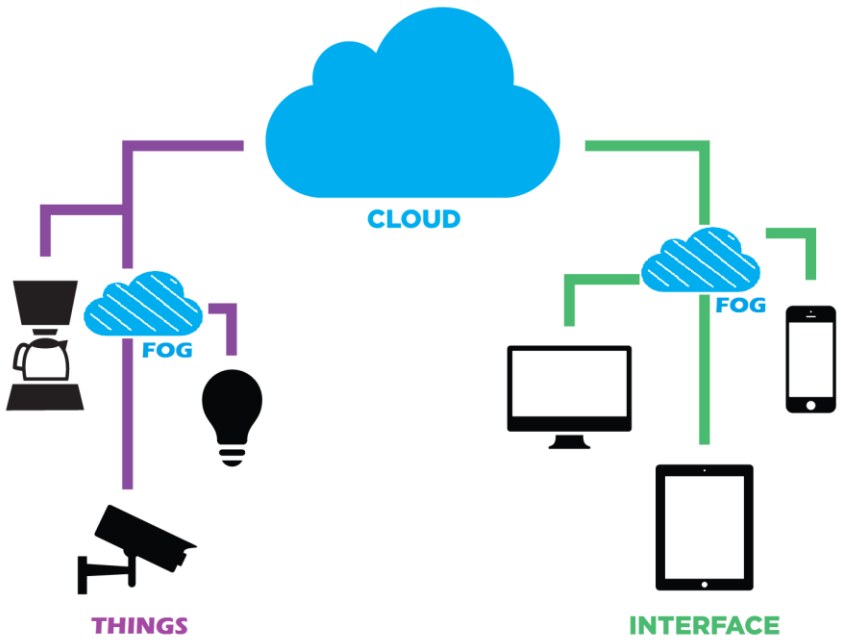
Fonte: Próprio Autor

2.3. Fog Computing

Um dos motivos pelo qual o cenário atual favorece o desenvolvimento da IIoT é a computação na nuvem (cloud computing). Cloud computing permite o armazenamento e análise da grande quantidade de dados que a comunicação entre as “coisas” gera, além de fornecer os requisitos de rede para uma comunicação em tempo real entre os dispositivos. No entanto, a nuvem está localizada na internet, que não garante uma qualidade de serviço (QoS), onde podem haver variações de velocidade e principalmente não garante segurança; quanto mais distante a fonte de dados se encontra do destino, por mais roteadores os dados irão trafegar e, portanto, menos segurança. [2]

A partir da necessidade de uma nuvem sem os problemas já citados, surge a névoa (Fog) (figura 5). A fog amplia a nuvem para aproximá-la aos dispositivos que produzem os dados, provendo assim uma maior agilidade, melhor segurança, uma melhor compreensão dos dados, com controle de privacidade e redução dos custos de operação. [8]

Figura 5: Ilustração de como funciona a Fog Computing



Fonte: http://iot-labs.com.my/2016/03/foggy-about-fog-computing/

2.4. Big Data

Dados grandes demais para serem administrados pela base de dados tradicionais são chamados de Big Data. Para obter o máximo de benefícios possíveis com a IIoT, as companhias devem analisar os dados de todas as fontes possíveis; o problema, no entanto, é que além de serem compostos por dados estruturados e não estruturados, as fontes serão os milhares de dispositivos conectados. [9]

Já existem serviços de nuvem capazes de gerir Big Data com capacidade ilimitada de armazenamento e tecnologia open source, como é o caso da Hadoop. Também existem ferramentas para análise desses dados, a exemplo da MapReduce, desenvolvida pela Google. [2]

2.5. Tecnologias de comunicação sem fio

2.5.1. Bluetooth 4.0

O Bluetooth é uma tecnologia de comunicação sem fio bastante conhecida de todos, presente em bilhões de dispositivos ao redor do mundo. Além disso, o novo Bluetooth 4.0 gasta pouca energia e foi projetado para rodar em dispositivos que trabalhem por pequenos intervalos de tempo. [10]

O Bluetooth 4.0 é ideal para certas aplicações no meio IIoT graças ao que é chamado de piconets, um grupo de dispositivos Bluetooth pareados que podem formar dinamicamente a medida que um dispositivo se aproxima ou se distancia dos outros. Piconets são formados por dois a oito dispositivos que se comunicam quando próximos. [2]

2.5.2. Zigbee

Zigbee é uma tecnologia sem fio global aberta que opera em uma área de até 70 metros. Usada principalmente em aplicações de controle e monitoramento de aplicativos que requerem baixa taxa de transferência e usam pouca energia,

o protocolo possibilita que os dispositivos se comuniquem em uma grande variedade de topologias de rede e as baterias podem durar vários anos. [11]

2.5.3. Wi-Fi Backscatter

A duração das baterias é um dos grandes problemas quando pensamos na utilização da tecnologia IIoT, no entanto o Retroespalhamento Wi-Fi pode ser a solução. A ideia dessa tecnologia é refletir as ondas de rádio e sua energia, produzidas por roteadores sem fio, televisores ou rádios para energizar dispositivos passivos sem bateria. [12]

2.5.4. RFID

A tecnologia de Radio-Frequency Identification (RFID) também entra na briga das comunicações sem fio por ser bastante popular na identificação de inventário, pessoas, objetos e animais pois suas tags podem ser anexadas em qualquer coisa. Além disso, não é necessário estar próximo para realizar leituras, as tags não precisam estar visíveis e, diferentemente de algumas tecnologias similares, centenas de tags RFID podem ser lidas ao mesmo tempo, como acontece nos sistemas de pedágio ao redor do mundo. [13]

2.6. IPv6

A versão mais usual do protocolo de Internet (IP) atualmente é a versão 4, mas isso deve mudar em breve. O IPv4 por si só não entrega segurança (necessita de TLS ou SSL) e possui baixo número de possíveis endereçamentos, tendo em vista que milhões de dispositivos estarão conectados com a rede e trocando informações; esses são alguns dos pontos importantes que essa versão do protocolo deixa a desejar. [2]

Outra característica do IPv6 é que o cabeçalho tem menos informação que o IPv4, isso possibilita que o roteamento, apesar carregarem pacotes

maiores, seja feita de forma mais rápida. A versão 6 desse protocolo possui uma melhor qualidade de serviço, maior extensibilidade e grande mobilidade. [14]

Apesar do IPv6 possuir numerosas vantagens sobre o IPv4 e ser a melhor opção para a IIoT, sua implementação ainda se dá de forma lenta. Ainda há grande receio por parte dos empresários do meio industrial e comercial na implementação da versão 6 do protocolo, pois vários dispositivos presentes na indústria, por serem mais antigos, não suportam o IPv6, e a troca desses aparelhos implica em gastos. Por esse motivo, a mudança para a versão 6 se dá de forma bastante lenta. [2][15]

2.7. Padrão publish/subscribe

O padrão publish/subscribe funciona da seguinte maneira: O aplicativo se cadastra em serviços publicados que estão interessados, dessa maneira o usuário receberá de forma imediata apenas aquilo que lhe é de interesse e ignorando os serviços irrelevantes, reduzindo o fluxo desnecessário de informação. Além disso, o broker não precisa de conexão constante com a internet, pois ele guarda a informação e a entrega quando o assinante ficar online. [2]

2.7.1. MQTT

O MQTT é um protocolo *publish/subscribe* focado na coleta de dados e é um protocolo *device-to-server*. O trabalho do MQTT se resume em coletar dados dos dispositivos e transportá-los ao servidor de coleta. O transporte dos dados se dá via TCP/IP, e apesar desse protocolo ser bastante confiável, sua utilização pode afetar a performance do MQTT. Esse protocolo é ideal para aplicações externas e com monitoramento remoto.

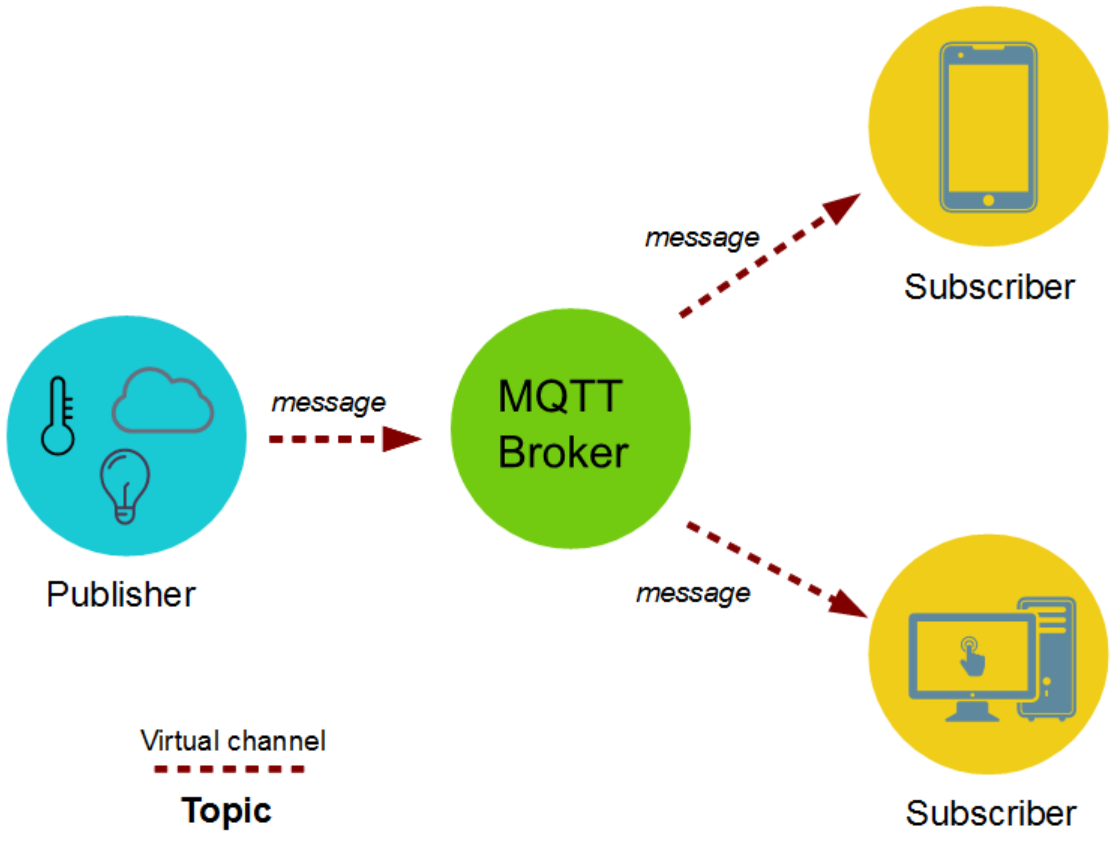
O publish/subscribe funciona da seguinte maneira: o cliente conecta-se a um servidor, que age como intermediário entre o publicador de mensagens e quem as recebe (broker), e publica uma mensagem em um determinado tópico, por exemplo, LIEC; apenas clientes ou aplicações que estejam inscritas nesse tópico LIEC receberão as mensagens publicadas nesse tópico; as aplicações

que não estiverem inscritas no tópico mencionado não receberão informação alguma (Figura 6).

O protocolo MQTT conta também com um recurso chamado sessão persistente. Caso o cliente opte por iniciar uma sessão persistente com o servidor, este salvará as informações relevantes, em casos de queda de conexão, e disponibilizará para o cliente no ato da reconexão, sem que seja necessário que o cliente realize uma nova assinatura.

O protocolo MQTT por conveniência foi escolhido para fornecer a comunicação entre o cliente (controle de acesso) e o servidor no sistema proposto graças a seu padrão publish/subscribe e a seu recurso de sessão persistente e por possuir três níveis de Quality of Service.

Figura 6: Etapas da comunicação MQTT



Fonte: https://www.survivingwithandroid.com/2016/10/mqtt-protocol-tutorial.html

2.7.2. XMPP

O protocolo XMPP (Extensible Messaging and Presence Protocol) foi desenvolvido para uso em mensagens instantâneas, detectar presença e permitir a conexão entre usuários para troca de mensagens [16]. XMPP é uma boa escolha quando usado em processos industriais que envolvem interface gerenciadas por humanos, mas não é recomendado em atividades que requeiram uma performance em tempo real, pois o XMPP se comunica utilizando HTTP sobre o TCP/IP, que combinado com o payload do XML, acaba sendo um processo demorado. [2]

2.7.3. DDS

O DDS (Data Distribution Service) é considerado um protocolo device-to-device e graças a essa característica, seu processamento de dados se dá em tempo real. Esse protocolo também oferece controle de qualidade de serviço detalhada e multicast.

Desenvolvido para sistemas em alta performance, o DDS é uma excelente escolha para aplicações em diversas das áreas na IIoT, tais como sistemas em manufatura, fazendas de eólicas, integração hospitalar e testes automotivos. [17]

3. Application Programming Interface (API)

Um API é uma interface programável para um aplicativo. Os componentes do API são:

- Application: Ferramentas, jogos, redes sociais e vários outros softwares que usamos no dia a dia.

- Programming: Trata-se do código que os engenheiros de software usam para criar os softwares para os dispositivos.

- Interface: Trata-se de um meio para que os usuários, aplicativos ou dispositivos possam interagir.

3.1. Web Services

Existem vários tipos de APIs, porém os mais utilizados atualmente nas Arquiteturas Orientadas a Serviço (SOA) e nos aplicativos para celular são SOAP e REST.

O Protocolo Simples de Arquitetura Aberta (SOAP) é um protocolo desenvolvido pela Microsoft com o intuito de substituir tecnologias antigas que não funcionavam bem com a internet. Graças a sua grande variedade de características, SOAP é altamente extensível, mas utiliza apenas as peças necessárias para realizar uma tarefa em particular. Como SOAP é intolerante a erros, muitos desenvolvedores Java, em particular, consideram que esse protocolo é de difícil uso. Em comparação ao SOAP, o Representational State Transfer (REST), provê uma alternativa leve para a web service API; ao invés de usar XML para fazer uma requisição, REST usa comandos basicos HTTP [2]. Na tabela 2, é feita uma comparação entre os protocolos SOAP e REST.

Tabela 2: Principais diferenças entre SOAP e REST

SOAP	**REST**
Linguagem, plataforma e transporte independentes	Não requer ferramentas caras para interagir com o web service
Trabalha bem em ambiente empresarial distribuído	Menor curva de aprendizado
Padronizado	Rápido
Manipulação de erro embutida	Eficiente
Automatizado quando usado com certas linguagens de produtos	Mais próximo à outras tecnologias web no que diz respeito a filosofia de design

Fonte: Próprio Autor.

4. Middlewares

Muitas vezes, no campo da computação distribuída, é necessário a comunicação entre programas que utilizam plataformas ou protocolos diferentes e, para que seja realizável essa troca de informações, faz-se necessário o uso de um software que se encontre entre o sistema operacional e os aplicativos executados no mesmo, esse software é chamado de middleware.[2]

A IIoT deu origem à necessidade de um novo tipo de middleware, projetado especificamente para as aplicações de Internet das Coisas no âmbito da Indústria. As novas plataformas middleware devem incluir: [18]

- Escalabilidade: Dezenas de milhares de dispositivos, todos tentando trocar informações entre si e entre o servidor central; para que isso aconteça, é imprescindível que haja comunicação entre as diversas tecnologias e protocolos ao mesmo tempo que seja abstraída a complexidade dos hardwares, softwares e protocolos de comunicação.
- Edge computing: A Industrial Internet of Things requer milhares de dispositivos com tempo real de comunicação e controle. Edge computing faz-se necessário pois diminui a latência da rede, entrega controle e monitoramento em tempo real.
- Arquiteturas de comunicação eficientes: Em muitas das aplicações onde a IIoT estará presente, não haverá uma boa conexão com a internet. Por isso, serão necessárias arquiteturas de comunicação e de rede mais eficientes.
- Computação Cognitiva: Quando se fala em IIoT, o que primeiro vem à mente é a capacidade que as aplicações terão, de prever uma falha antes mesmo que ela aconteça. A base para a análise preditiva é a computação cognitiva, computadores que imitam o modo como a mente humana trabalha.

No mercado, já existem diversas soluções open source para IIoT middlewares; dentre eles encontram-se as seguintes:

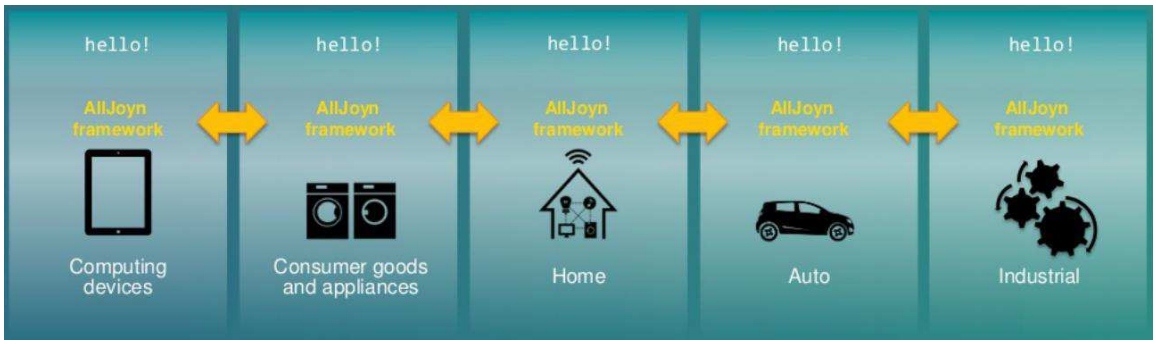
- Kaa: Fornece as ferramentas necessárias para construir soluções completas para a Industrial Internet. Kaa também fornece meios para monitorar e gerenciar cada dispositivo remoto (Figura 7). [19]

Figura 7: Proposta da Kaa middleware

Fonte: www.kaaproject.org

- OpenIoT: Plataforma middleware open source para implementação e integração de soluções IoT. OpenIoT, é capaz de conectar e então coletar e processar dados de praticamente qualquer transdutor, independentemente de seu protocolo. [20]
- Alljoyn: Trata-se de uma plataforma que facilita a comunicação entre dispositivos independentemente de seu protocolo, de sua manufatura ou da sua camada de transporte (Figura 8). [21]

Figura 8: Proposta da AllJoyn middleware



Fonte: https://www.slideshare.net/AllSeenAlliance/intro-to-alliance2252015

- Mango: Graças a sua tradição em M2M, controle industrial e ambientes industriais SCADA, essa é uma das plataformas IoT mais populares. Fácil implantação, baixo consumo de energia e pode hospedar milhares de dispositivos. [22]

Apesar de bastante popular, nem sempre a plataforma open source é a melhor opção em certos ambientes industriais. Abaixo estão listadas as plataformas middleware para IoT mais populares, mas que precisam ser compradas. [18]

- ThinkWorx
- Oracle Fusion
- IBM Bluemix

5. Segurança

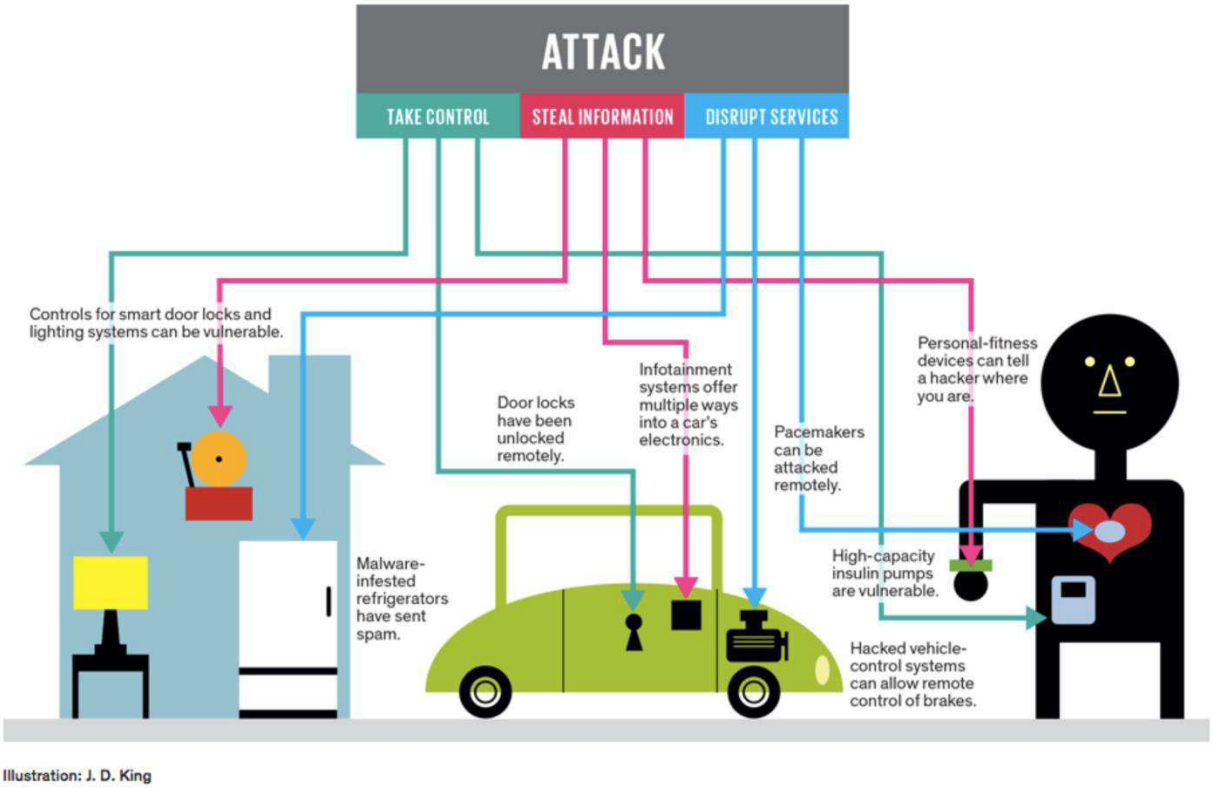
A principal preocupação associada ao desenvolvimento e implementação de aplicações IoT é a segurança (figura 9). Dificilmente os sistemas industriais rodam em Windows ou Linux, ao invés disso, sua grande maioria roda em pequenos sistemas operacionais conectados através de protocolos diferentes do IP e topologia de barramento serial. A ameaça de ataques cibernéticos que podem prejudicar todo o processo indústrial ou até mesmo hackers buscando obter informações privilegiadas para extorsão existe, e algumas companhias já sofreram ataques, segundo Marina Krytofil. [2]

A tecnologia IoT é implementada pelas grandes indústrias mesmo sem a garantia de segurança adequada que a tecnologia deve ter. No ano de 2015, pesquisadores da Rapid7 revelaram que um grande número de babás eletrônicas que usavam a tecnologia IoT possuíam vulnerabilidades críticas, portanto hackers poderiam invadi-las para permitir o acesso a outras pessoas, controlar o monitor, etc. [23]

Outro episódio parecido aconteceu em 2015, quando dois pesquisadores de segurança, Charlie Miller e Chris Valasek foram capazes de hackear um Jeep Cherokee remotamente, primeiro tomando controle do sistema de entretenimento do veículo, e então desabilitando o acelerador e o freio do mesmo. Esse episódio serviu para que toda a indústria IoT acordasse para a importância da segurança na Internet of Things. [24]

Para que a tecnologia IoT possa ser vastamente implementada, é necessário que o problema da segurança seja totalmente erradicado; as grandes empresas não podem correr o risco de seus concorrentes obterem informações privilegiadas vazadas ou adquiridas via ataques cibernéticos, bem como não são aceitáveis as extorsões de alguns hackers. Enquanto o problema da segurança não for resolvido, dificilmente a IoT será vastamente implantada no mercado.

Figura 9: A segurança na IIoT



Fonte: https://www.linkedin.com/pulse/consumers-iot-early-adoption-can-slippery-slope-roger-attick/

6. Sistema de controle de acesso distribuído

Nesse tópico será descrito o sistema desenvolvido neste trabalho. Nas subseções a seguir estarão detalhados o objetivo do sistema, os hardwares e softwares utilizados para seu desenvolvimento e algumas dificuldades que enfrentei no processo.

6.1. Objetivos do sistema

O objetivo do sistema de automação desenvolvido neste trabalho de conclusão de curso é automatizar o controle de acesso da sala de IIoT do LIEC. Até então, o acesso à sala de IIoT é feito através de uma senha de quatro dígitos e o gerenciamento de credenciais é no próprio aparelho, de forma local.

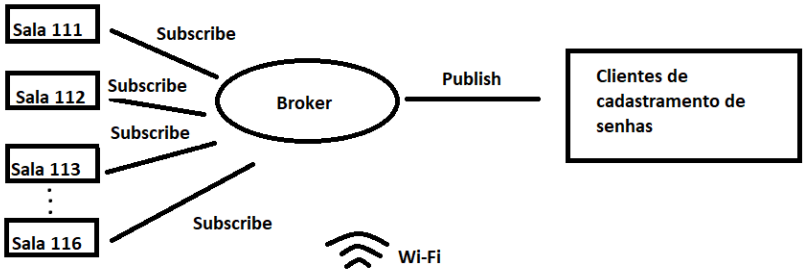
O novo sistema funciona a partir da leitura das tags NFC dos usuários. Por meio do protocolo de comunicação MQTT, o Arduino UNO recebe uma lista com os IDs das tags credenciadas. Após recebida a lista, o leitor NFC é capaz de reconhecer quais tags podem ou não acessar a sala. O gerenciamento das tags pode ser feito por qualquer aplicativo que reconheça o padrão MQTT e realize inscrição e publicação no tópico referente à sala.

6.2. Funcionamento do Sistema

O novo sistema, como pode ser visto na figura 10, consiste no envio dos IDs para o Arduino. O cliente de cadastramento publica os números identificadores das tags no tópico referente à sala, o broker faz o papel de intermediar essa mensagem e enviar através do Wi-Fi ao NodeMCU, que está inscrito no mesmo tópico que o cliente enviou a mensagem, que por sua vez passa, através da comunicação serial, essa informação ao Arduino, que guarda esses identificadores em sua memória. Dessa maneira, quando uma tag é lida pelo sensor NFC e enviada para o Arduino, esse verifica se o número identificador lido consta em sua memória e libera o acesso caso receba uma

confirmação. O código completo implementado no Arduino Uno e no NodeMCU constam no fim deste trabalho, nos Apêndices A e B.

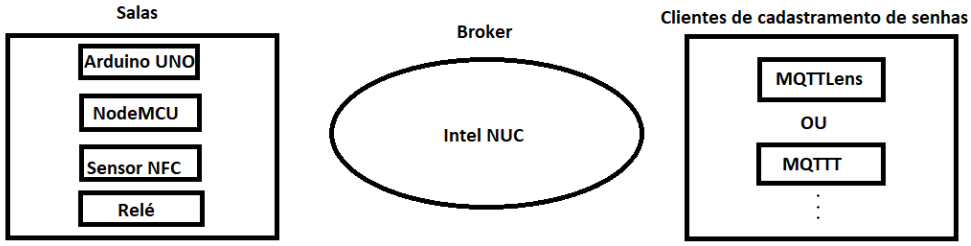
Figura 10: Comunicação do sistema



Fonte: Próprio Autor

Na figura 11, é mostrado mais detalhadamente em que consistem as salas, broker e clientes de cadastramento de senhas que foram apresentados na figura 10.

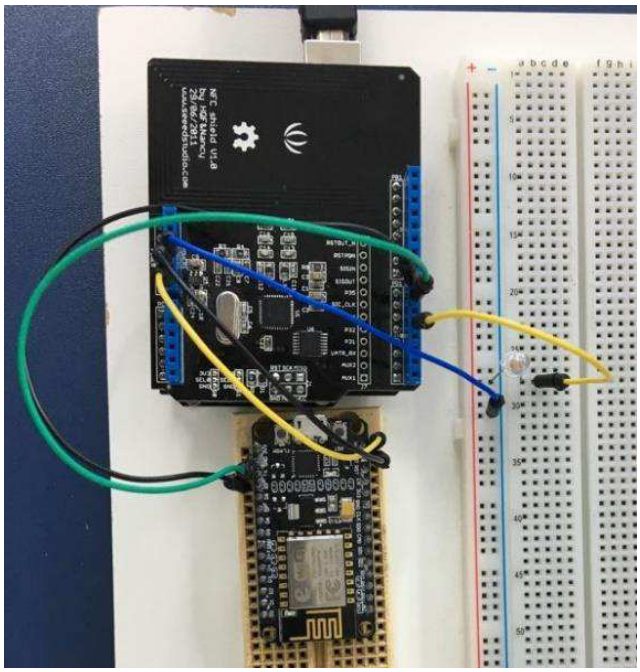
Figura 11: Detalhamento do sistema



Fonte: Próprio Autor

Abaixo, na figura 12, é possível ver as ligações feitas entre o Arduino Uno, o Sensor NFC e o NodeMCU.

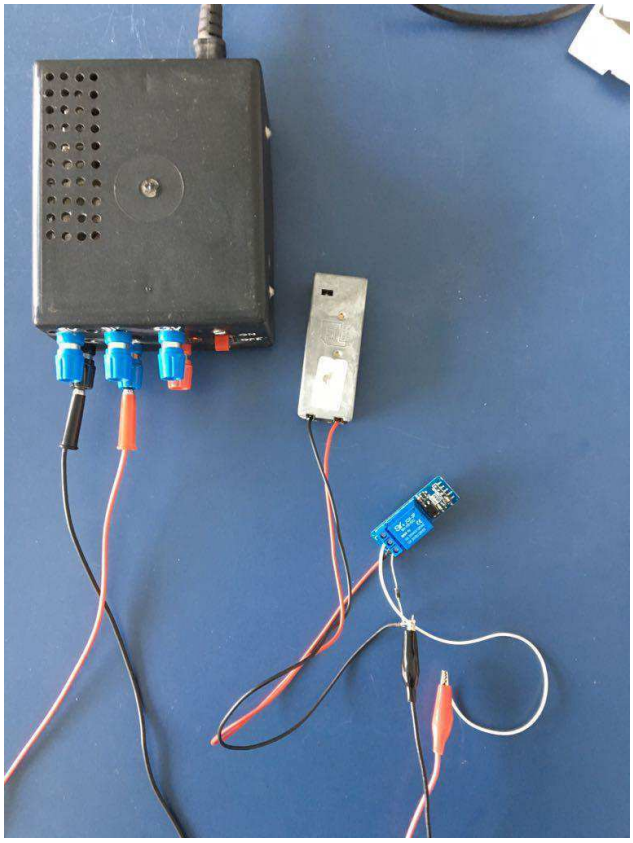
Figura 12: Conexões do UNO, leitor NFC e NodeMCU



Fonte: Próprio autor

A ligação entre o relé e a fechadura eletrônica está ilustrada abaixo, na figura 13.

Figura 13: Ligação entre o relé e a fechadura eletrônica

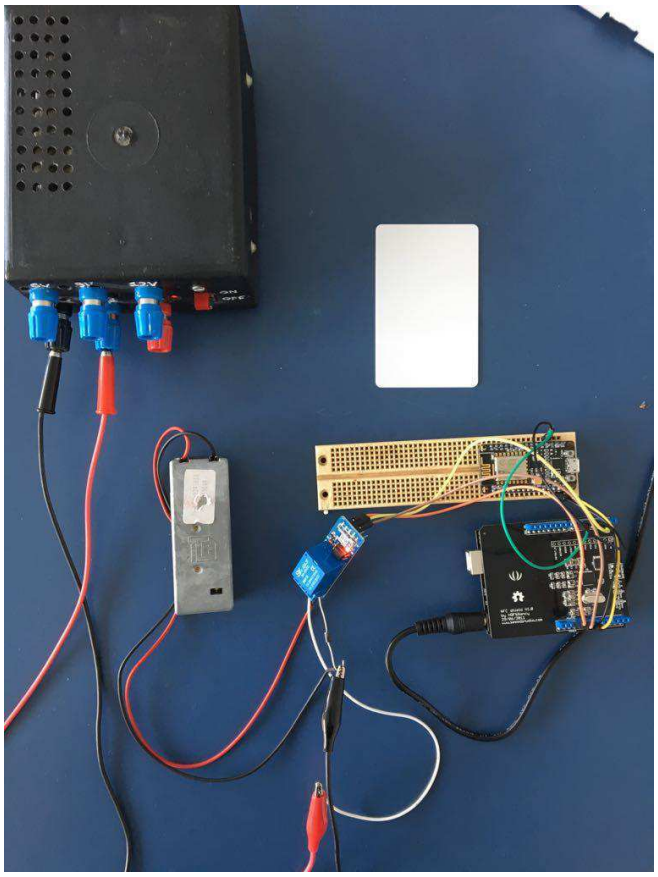


Fonte: Próprio autor

Na figura 14, é possível visualizar as ligações que foram feitas entre o Arduino Uno, o NodeMCU, o sensor NFC, o relé e a fechadura eletrônica. Foi utilizado também um diodo entre a fechadura e o relé, de forma que impeça que corrente faça o caminho inverso, podendo danificar o sistema.

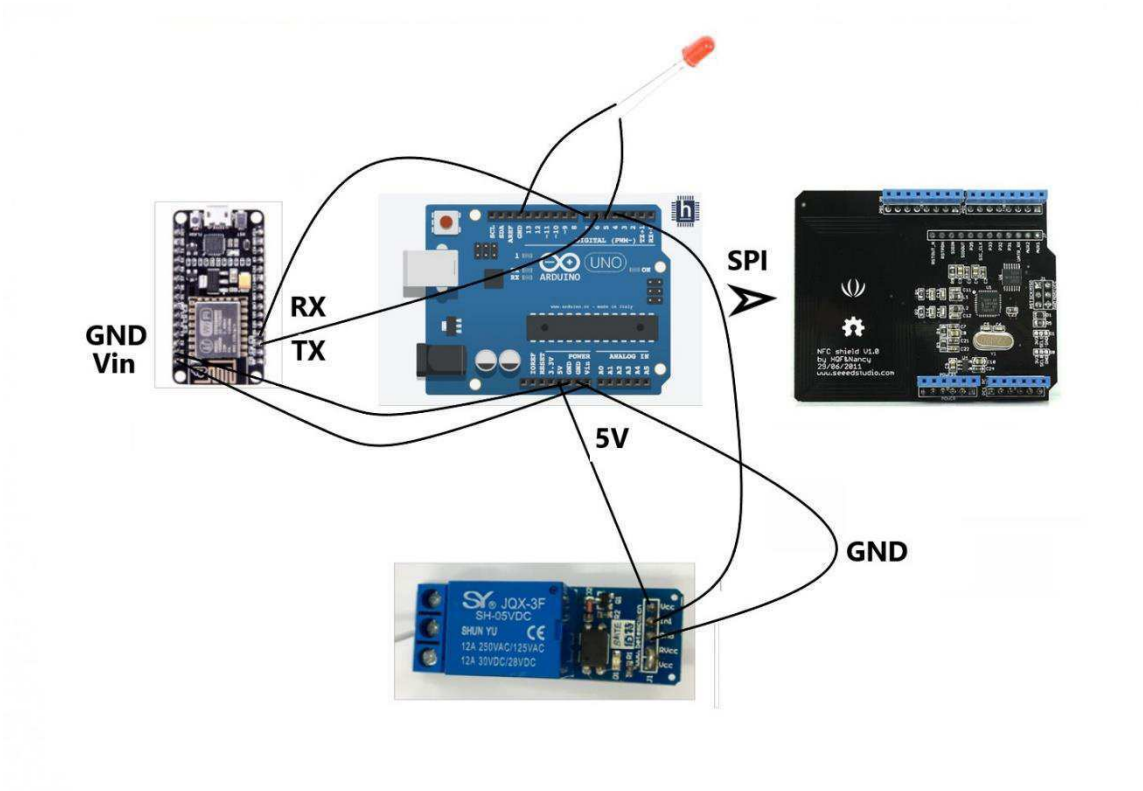
É possível ver na figura 15 as ligações vistas na figura 14 de forma mais detalhada, ilustrando melhor as ligações feitas. De forma que, no Arduino UNO, os pinos D6 e D7 foram configurados como RX e TX, respectivamente, o pino D4 gera a saída do sinal de pulso que vai para o relé e o pino D5 gera um sinal para o LED.

Figura 14: Ligação completa dos elementos do sistema



Fonte: Próprio Autor

Figura 15: Ligações feitas no sistema



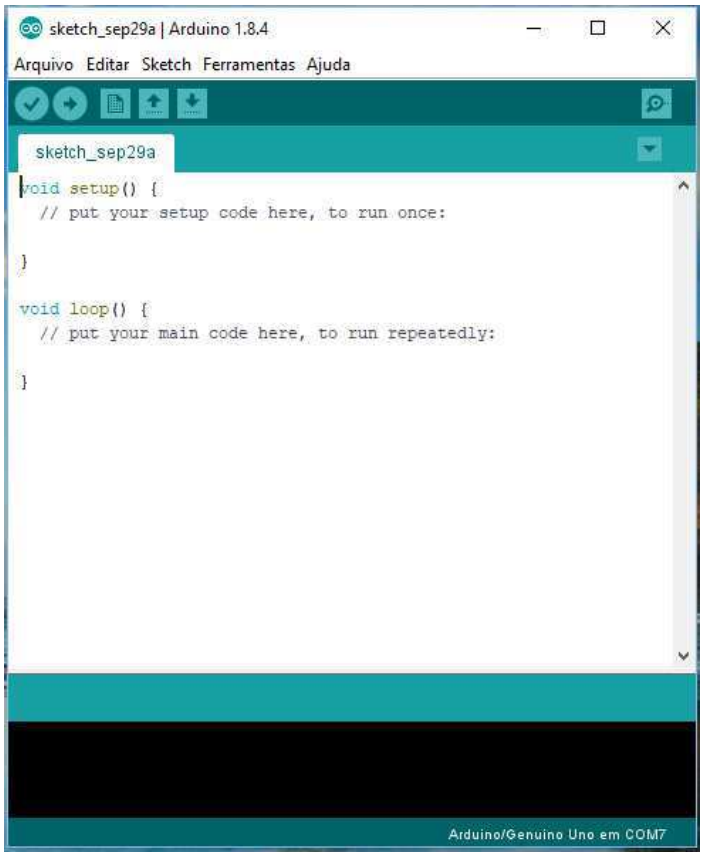
Fonte: Próprio autor.

6.3. Softwares utilizados

- Arduino Integrated Development Enviroment (IDE)

O Arduino IDE é um ambiente de programação open-source desenvolvido pela Arduino que possibilita o desenvolvimento de códigos para as placas da sua empresa e algumas outras que forem compatíveis, como é o caso do ESP8266 utilizado neste projeto. O IDE da Arduino foi utilizado para compilar e carregar o programa desenvolvido para o Arduino UNO e o NodeMCU. O ambiente de programação do Arduino encontra-se ilustrado na figura 16.

Figura 16: Arduino IDE



Fonte: Software no computador do próprio autor.

Na IDE do Arduino, foram utilizadas algumas bibliotecas desenvolvidas por outros usuários, que possibilitaram a implementação do sistema. As bibliotecas utilizadas foram:

- PubSubClient: Biblioteca criada por Nick O'Leary, tem como objetivo prover a comunicação publish/subscribe entre clientes e um servidor que suporte o padrão MQTT. Os hardwares compatíveis com essa biblioteca são: Arduino Ethernet, Arduino Shield, Arduino YUN, Arduino WiFi Shield, Sparkfun WiFly Shield, Intel Galileo/Edison e ESP8266. [25]

- PN532_SPI: Biblioteca desenvolvida pela SeeedStudio, tem como objetivo integrar o NFC Shield desenvolvido pela própria empresa vários hardwares compatíveis da Arduino.

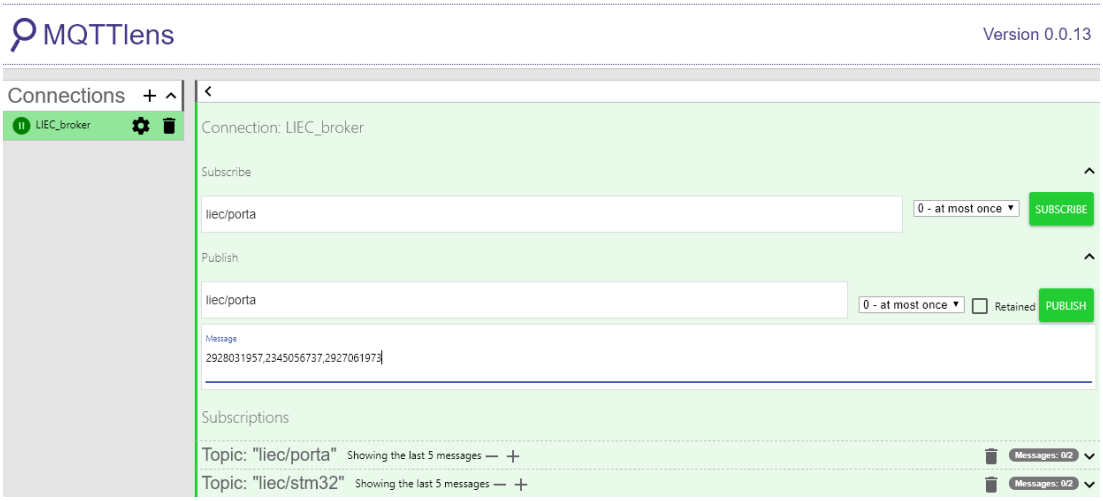
- Eclipse Mosquitto – MQTT broker

Eclipse Mosquitto é um servidor de mensagens open-source que implementa a versão 3.1 e 3.1.1 do protocolo MQTT, o Mosquitto provê um método leve de carregar mensagens usando o modelo publish/subscribe, tornando-o adequado para o uso em aplicações que usem a Internet das Coisas [26]. Neste projeto, o mosquitto foi embarcado no Intel NUC, um mini PC, para servir como broker na rede de comunicação MQTT criada.

- MQTTLens

Trata-se de um aplicativo disponível na chrome web store, que se conecta a um broker MQTT e é capaz de se inscrever e publicar mensagens em tópicos MQTT. O aplicativo MQTTLens é ilustrado abaixo, na figura 17. [27]

Figura 17: MQTTLens App

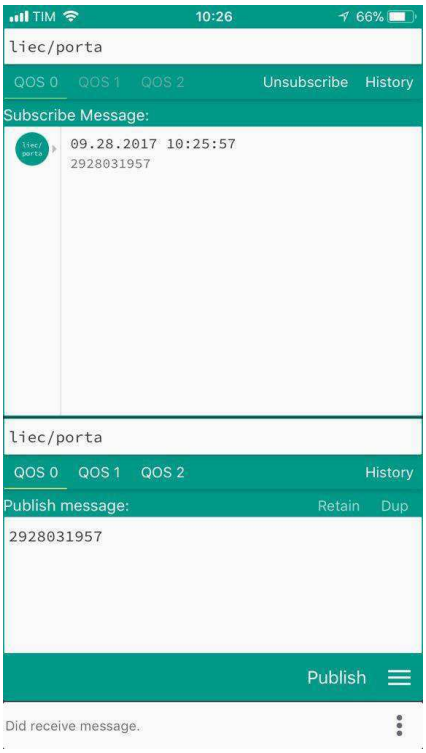


Fonte: Aplicativo no computador do próprio autor.

- MQTTT

O MQTTT é uma ferramenta desenvolvida por Chenhsin Wong, disponível na Apple Store, usada para enviar e receber mensagens MQTT [28]. Na figura 18, é mostrado o aplicativo MQTTT.

Figura 18: MQTTT App



Fonte: Aplicativo no celular do próprio autor.

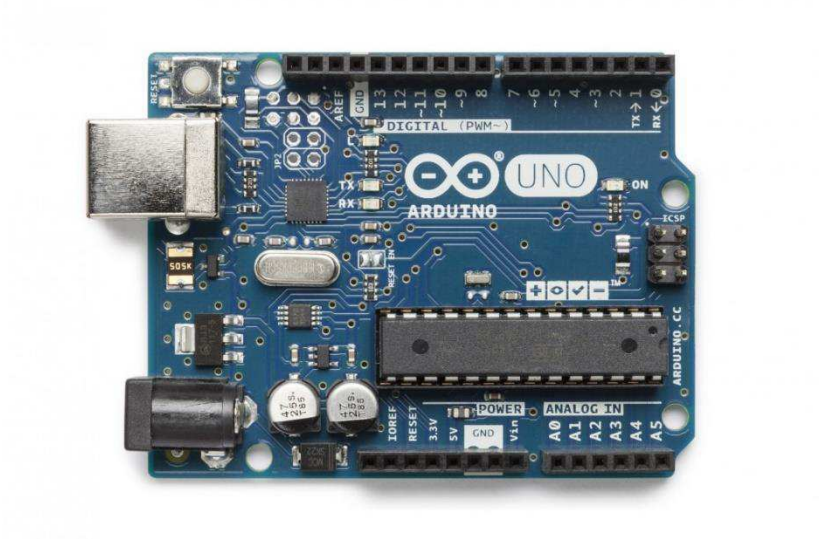
6.4. Hardwares utilizados

- Arduino UNO

O Arduino UNO (figura 19) é uma placa de microcontrolador baseada no chip ATmega328P. O UNO possui [29]:

- 14 pinos de entrada e saída digitais
- Seis entradas analógicas
- Um cristal de quartzo de 16MHz
- Uma conexão USB
- Uma tomada de força
- Um cabeçalho ICSP (In-circuit serial programming)
- Um botão reset

Figura 19: Placa do Arduino UNO

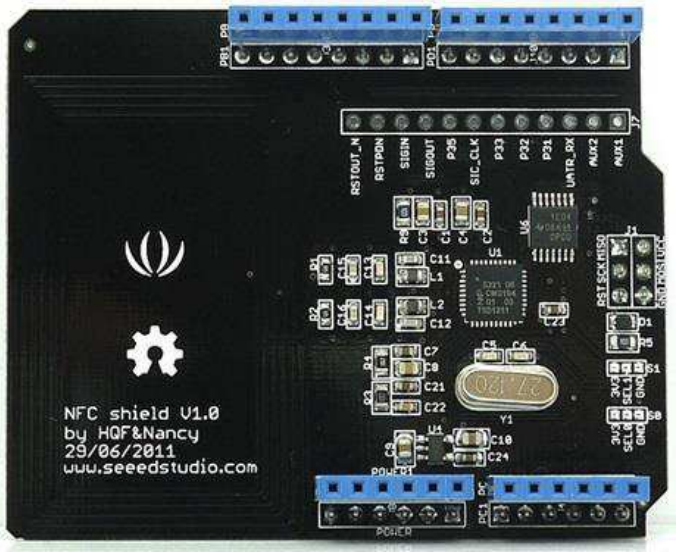


Fonte: https://store.arduino.cc/usa/arduino-uno-rev3

- NFC Shield v1.0

O NFC Shield (figura 20) é uma interface NFC de comunicação desenvolvida para Arduino pela SeeedStudio. Possui interface ISP (Interface Segregation Principle), uma antena PCB (Printed Circuit Board) embutida e suporta uma tensão de operação de 3.3V e 5V [30].

Figura 20: NFC Shield v1.0



Fonte: https://www.seeedstudio.com/NFC-Shield-p-916.html

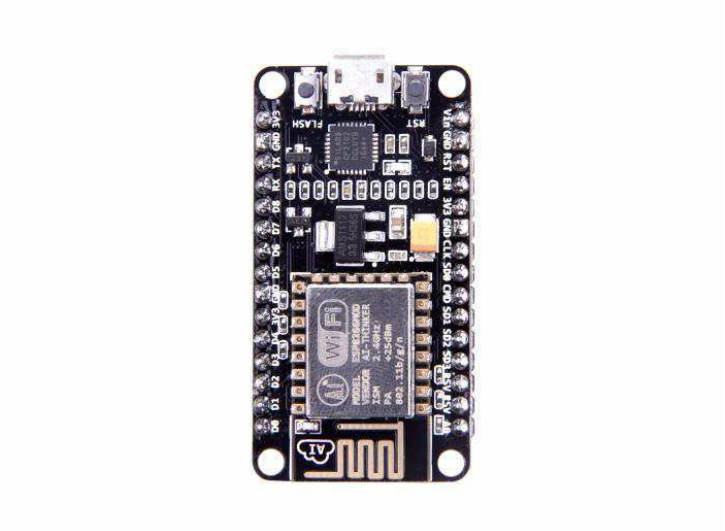
- NodeMCU

O NodeMCU (figura 21) é uma plataforma IoT open source. Possui um firmware que roda no ESP8266 e um hardware baseado no módulo ESP-12E. O NodeMCU é open source, programável, de baixo custo, simples, inteligente e possui conexão Wi-Fi.

O módulo ESP-12E presente no NodeMCU possui as seguintes especificações: [31]

- Wireless padrão 802.11 b/g/n
- Antena embutida
- Modos de operação: STA/AP/STA+AP
- Segurança WEP, WPA, TKIP, AES
- Protocolo TCP/IP embutido
- Portas GPIO: 11
- Tensão de operação: 3,3V
- Taxa de transferência: 110-460800bps
- Conversor analógico digital (ADC)
- Distrancia entre pinos: 2mm
- Dimensoes: 24 x 16 x 3,2mm

Figura 21: Placa NodeMCU



Fonte: https://www.seeedstudio.com/NodeMCU-v2-Lua-based-ESP8266-development-kit-p-2415.html

- **Intel NUC**

Mini PC (Personal Computer) de 4 x 4 polegadas produzido pela Intel e que foi utilizado para embarcar o broker mosquitto necessário para a comunicação MQTT. As especificações completas do intel NUC (figura 23) podem ser encontradas no seguinte link: https://www.quietpc.com/intel-nuc.

- **Placa de relé**

Na figura 22, é ilustrado o módulo de relé utilizado no projeto, suas especificações são as seguintes: [32]

- Tipo: Digital
- Bobina: 5VDC 75mA
- Carga nominal do relé: 12A, 125VAC, 7A 250VAC
- Carga nomial do módulo: 10A
- Tempo de acionamento de contato: 10ms

Figura 22: Módulo relé

Fonte: Próprio autor

Figura 23: Mini PC Intel NUC

Fonte: https://www.quietpc.com/intel-nuc

7. Considerações Finais

Após todo o estudo realizado sobre o estado da arte da Industrial Internet of Things e aprendizado sobre como funciona essa nova tecnologia, foi possível concluir que a IoT é uma alternativa indispensável para o setor industrial e inevitavelmente será uma realidade em todo o mundo em alguns poucos anos.

O grande desafio hoje para a implementação da IIoT é justamente a segurança, que ainda é bastante precária, porém como se trata de uma tecnologia revolucionaria para a época, tenho certeza que não demorará até que esse problema seja resolvido e a IIoT esteja implantada em todo o cenário industrial.

7.1. Trabalhos Futuros

Até o presente momento, o tópico e o broker não possuem proteção e qualquer usuário que possua o número IP da conexão do broker pode entrar no sistema e assim habilitar sua entrada na sala. A segurança para se conectar com o broker está sendo desenvolvida e deve ser aplicada em breve.

Para o futuro, é interessante também elaborar uma placa de circuito impressa, implementar um cliente MQTT customizado para cadastrar os usuários de uma forma mais amigável, testar o emparelhamento com uma tag dinâmica NFC num smartphone e expandir o sistema para as outras salas do prédio.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] O surgimento da Internet. Disponível em: < https://www.maxwell.vrac.puc-rio.br/9888/9888_4.PDF >. Acesso em: 29/09/2017.

[2] GILCHRIST, Alasdair. **Industry 4.0: The industrial Internet of Things**. Bangken: Apress, 2016.

[3] The Internet of Things for Health Care: A Comprehensive Survey. Disponível em: < http://ieeexplore.ieee.org/document/7113786/#full-text-section >. Acesso em: 29/09/2017.

[4] IIoT use cases in the oil and gas industry. Disponível em: < https://www.rcrwireless.com/20160720/internet-of-things/use-cases-iot-oil-gas-tag31-tag99 >. Acesso em: 29/09/2017.

[5] IoT in Oil & Gas: 5 Real World Use Cases. Disponível em: < https://www.linkedin.com/pulse/iot-oil-gas-5-real-world-use-cases-jonathan-miller/ >. Acesso em: 29/09/2017.

[6] IoT vs M2M. Disponível em : < https://www.pubnub.com/blog/2015-01-02-iot-vs-m2m-understanding-difference/ >. Acesso em: 20/09/2017.

[7] IoT and M2M, what's the difference?. Disponível em : < http://www.incognito.com/blog/iot-and-m2m-whats-the-difference/ >. Acesso em: 20/09/2017.

[8] Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are. Disponível em: < https://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf >. Acesso em: 29/09/2017.

[9] Big Data. Disponível em: < http://searchcloudcomputing.techtarget.com/definition/big-data-Big-Data >. Acesso em: 29/09/2017.

[10] Bluetooth 4.0: What is it, and does it matter? Disponível em: < https://www.cnet.com/news/bluetooth-4-0-what-is-it-and-does-it-matter/ >. Acesso em: 29/09/2017.

[11] Zigbee Wireless Standart. Disponível em: < https://www.digi.com/resources/standards-and-technologies/rfmodems/zigbee-wireless-standard >. Acesso em: 29/09/2017.

[12] Wi-Fi sem baterias é nova esperança da Internet das Coisas. Disponível em: < http://www.inovacaotecnologica.com.br/noticias/noticia.php?artigo=wi-fi-sem-baterias#.Wc6kTWhSzIU >. Acesso em: 29/09/2017.

[13] Funcionamento da RFID. Disponível em: < http://saladaautomacao.com.br/funcionamento-da-rfid/ >. Acesso em: 29/09/2017**.**

[14] What is The Difference Between IPv6 and IPv4?. Disponível em: < http://www.webopedia.com/DidYouKnow/Internet/ipv6_ipv4_difference.html >. Acesso em: 29/09/2017

[15] IPv4 & IPv6: A Short Guide. Disponível em: < http://mashable.com/2011/02/03/ipv4-ipv6-guide/#KbazQV_dPOql >. Acesso em: 29/09/2017.

[16] Na Overview of XMPP. Disponível em: < https://xmpp.org/about/technology-overview.html >. Acesso em:08/10/2017.

[17] Data Distribution Service. Disponível em: < http://www.prismtech.com/vortex/technologies/data-distribution-service >. Acesso em: 29/09/2017.

[18] Automation.com. **2017:State of The IIoT**: Key Trends and predictions for the industrial internet of Things.

[19] Kaa middleware. Disponível em: < www.kaaproject.org >. Acesso em: 22/09/2017.

[20] OpenIoT middleware. Disponível em: < https://github.com/OpenIotOrg/openiot >. Acesso em:22/09/2017.

[21] Alljoyn middleware. Disponível em: < https://allseenalliance.org/ >. Acesso em: 22/09/2017.

[22] Mango middleware. Disponível em: < https://github.com/paulbellamy/mango >. Acesso em: 22/09/2017.

[23] Why IoT Security is so critical. Disponível em: < https://techcrunch.com/2015/10/24/why-iot-security-is-so-critical/ >. Acesso em: 22/09/2017.

[24] Five Lessons On The 'Security Of Things' From The Jeep Cherokee Hack. Disponível em: < https://www.forbes.com/sites/johnvillasenor/2015/07/27/five-lessons-on-the-security-of-things-from-the-jeep-cherokee-hack/#571ae8be692a >. Acesso em: 22/09/2017.

[25] Arduino Client for MQTT. Disponível em: < https://pubsubclient.knolleary.net/ >. Acesso em: 29/09/2017.

[26] Eclipse Mosquitto. Disponível em: < https://mosquitto.org/ >. Acesso em: 29/09/2017.

[27] MQTTLens. Disponível em: < https://chrome.google.com/webstore/detail/mqttlens/hemojaaeigabkbcookmlgmdigohjobjm?hl=en >. Acesso em: 29/09/2017.

[28] MQTTT. Disponível em: < https://itunes.apple.com/br/app/mqttt/id1217080708?mt=8 >. Acesso em: 29/09/2017.

[29] Arduino UNO Rev3. Disponível em: < https://store.arduino.cc/usa/arduino-uno-rev3 >. Acesso em: 29/09/2017.

[30] NFC Shield. Disponível em: < https://www.seeedstudio.com/NFC-Shield-p-916.html >. Acesso em: 29/09/2017.

[31] Módulo WiFi ESP8266 ESP-12E. Disponível em: < https://www.filipeflop.com/produto/modulo-wifi-esp8266-esp-12e/ >. Acesso em: 30/09/2017.

[32] Módulo relé 5V – 2 canais. Disponível em: < http://www.baudaeletronica.com.br/modulo-rele-5v-2-canais.html >. Acesso em: 02/09/2017.

Apêndice A – Código Fonte NodeMCU

```
#include <ESP8266WiFi.h>

#include <PubSubClient.h>

#include <string.h>


const char* ssid = "LIEC_Wireless2";

const char* password = "0987ABCDEF";

int keyIndex = 0;

char topic = 'E';

int i=0;

IPAddress broker(150, 165, 52, 98);  //raspbe pi private IP

WiFiClient  wifiClient;

PubSubClient client(wifiClient);

char payload[1];


void setup() {

  Serial.begin(9600);

  delay(10);

  client.setServer(broker, 1883);

  client.setCallback(callback);
```

```
  // Connect to WiFi network
  WiFi.begin(ssid,password);
  while (WiFi.status() != WL_CONNECTED){
    delay(500);
    Serial.println("Connected to WIFI");
  }
 WiFi.mode(WIFI_STA);
}
void loop() {
  if (!client.connected()) {
    reconnect();
  }
    client.loop();
}
void callback(char* topic, byte* payload, unsigned int length) {
  //Serial.print("Message arrived [");
  //Serial.print(topic);
  //  Serial.print("] ");
  for (int i=0;i<length;i++) {
     Serial.print((char)payload[i]);
  }
  Serial.print("\n");
}
```

```
void reconnect() {
  // Loop until we're reconnected
  while (!client.connected()) {
    //Serial.print("Attempting MQTT connection...");
    // Attempt to connect
    if (client.connect("stm32Client")) {
      Serial.println("connected to MQTT");
      // Once connected, publish an announcement...
      client.publish("liec/stm32","hello, from stm32");
      client.subscribe("liec/porta");
      // ... and resubscribe
    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      // Wait 5 seconds before retrying
      delay(5000);
    }
  }
}
```

Apêndice B – Código Fonte Arduino UNO + NFC Shield

```
#include <PN532.h>
#include <SPI.h>
#include <SoftwareSerial.h>

#define PN532_CS 10
PN532 nfc(PN532_CS);
#define  NFC_DEMO_DEBUG 1
#define numusuarios 10
uint32_t listausuarios[numusuarios];
#define PULSO 4
#define LED 5
SoftwareSerial espSerial(6,7); //RX = 6, TX = 7

void setup(void) {
  pinMode(PULSO, OUTPUT);
  pinMode(LED, OUTPUT);

#ifdef NFC_DEMO_DEBUG
  Serial.begin(9600);
```

```
  Serial.println("Hello!");
#endif
  espSerial.begin(9600);
  nfc.begin();
  uint32_t versiondata = nfc.getFirmwareVersion();

  if (! versiondata) {
#ifdef NFC_DEMO_DEBUG
    Serial.print("Didn't find PN53x board");
#endif
    while (1); // halt
  }

#ifdef NFC_DEMO_DEBUG
  // Got ok data, print it out!
  //Serial.print("Found chip PN5");
  //Serial.println((versiondata>>24) & 0xFF, HEX);
  //Serial.print("Firmware ver. ");
  //Serial.print((versiondata>>16) & 0xFF, DEC);
  //Serial.print('.');
  //Serial.println((versiondata>>8) & 0xFF, DEC);
  //Serial.print("Supports ");
  //Serial.println(versiondata & 0xFF, HEX);
```

```
#endif
  // configure board to read RFID tags and cards
  nfc.SAMConfig();

 }
void loop(void) {

  while (espSerial.available()) {

    listausuarios[0] = espSerial.parseInt();
    listausuarios[1] = espSerial.parseInt();
    listausuarios[2] = espSerial.parseInt();
    listausuarios[3] = espSerial.parseInt();
    listausuarios[4] = espSerial.parseInt();
    listausuarios[5] = espSerial.parseInt();
    listausuarios[6] = espSerial.parseInt();
    listausuarios[7] = espSerial.parseInt();
    listausuarios[8] = espSerial.parseInt();
    listausuarios[9] = espSerial.parseInt();
    break;
  }
  uint32_t id;
  // look for MiFare type cards
```

```
  id = nfc.readPassiveTargetID(PN532_MIFARE_ISO14443A);
  if (id != 0) {
    #ifdef NFC_DEMO_DEBUG
    Serial.print("   Read card #");
    Serial.println(id);
    #endif

if((id==listausuarios[0])||(id==listausuarios[1])||(id==listausuarios[2])||(id==listausuarios[3])||(id==listausuarios[4])||(id==listausuarios[5])||(id==listausuarios[6])||(id==listausuarios[7])||(id==listausuarios[8])||(id==listausuarios[9])) {
      digitalWrite(LED,HIGH);
      digitalWrite(PULSO,HIGH);
      delay(200);
      digitalWrite(PULSO,LOW);
      delay(1000);
      digitalWrite(LED,LOW);
  }
}
}
```