INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PIAUÍ
IFPI - CAMPUS PICOS
TECNÓLOGO EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

JOSÉ ALAN DOS SANTOS ROCHA

MONITORAMENTO CLIMÁTICO DO *DATA CENTER* DA JUSTIÇA FEDERAL DE PICOS ATRAVÉS DA IOT

PICOS-PI
2018

FICHA CATALOGRÁFICA
Sistema de Bibliotecas
Gerada automaticamente com dados fornecidos pelo(a) autor(a)

Rocha, José Alan dos Santos
R673m MONITORAMENTO CLIMÁTICO DO DATA CENTER DA JUSTIÇA FEDERAL DE PICOS ATRAVÉS DA IOT / José Alan dos Santos Rocha - 2018.
63 p. : il. p&b.

Trabalho de conclusão de curso (Tecnologia) - Instituto Federal de Educação, Ciência e Tecnologia do Piauí, Campus Picos, Tecnologia em Análise e Desenvolvimento de Sistemas, 2018.

Orientador : Prof Esp. Juciê Xavier da Silva.
Coorientador : Prof Me. Rogério Figueredo.

1. IOT. 2. SERVIDOR. 3. TEMPERATURA. 4. UMIDADE . 5. ARDUINO. I.Título.

CDD 004

JOSÉ ALAN DOS SANTOS ROCHA

MONITORAMENTO CLIMÁTICO DO *DATA CENTER* DA JUSTIÇA FEDERAL DE PICOS ATRAVÉS DA IOT

Trabalho de conclusão de curso apresentado como exigência final para obtenção do diploma do Curso Tecnólogo em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Piauí, Campus-Picos.

Orientador**:** Prof. Esp. Juciê Xavier da Silva.

PICOS-PI
2018

JOSÉ ALAN DOS SANTOS ROCHA

MONITORAMENTO CLIMÁTICO DO *DATA CENTER* DA JUSTIÇA FEDERAL DE PICOS ATRAVÉS DA IOT

Trabalho de conclusão de curso apresentado como exigência final para obtenção do diploma do Curso Tecnólogo em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Piauí - IFPI. Campus Picos.

Orientador: Prof. Esp. Juciê Xavier da Silva.

Aprovado pela banca examinadora em: 03 / Maio de 2018.

BANCA EXAMINADORA:

Prof. Esp. Juciê Xavier da Silva
Orientador
(IFPI)

Prof. Ms. Rogério Figueiredo de Sousa
(IFPI)

Prof. Esp. Vicente Medeiros Carvalho
(IFPI)

A minha esposa Eduarda, meus pais Hilda e Lourenço, minha família, amigos e professores.

AGRADECIMENTOS

Agradeço primeiramente à Deus, pela força que me deu quando mais precisava, a meus pais Lourenço e Hilda, por me darem todo o apoio necessário e sempre acreditarem em mim, à minha esposa Eduarda por sempre estar ao meu lado nos momentos mais difíceis.

Agradeço ao meu Professor e Orientador Esp. Juciê Xavier pelos ensinamentos e pela confiança que me passou. Ao Professor Ms. Rogério Figueiredo pela paciência, que foi muita, e pela dedicação durante todo o curso.

Agradeço ao meu amigo Cleomar que me deu a primeira oportunidade de trabalhar nessa área de T.I e sempre me incentivou a seguir os estudos. Aos meus amigos: Ítalo, Tulio, Platini (O Bombeiro), Leoberson e Dutra por todo o apoio nessa jornada. Agradeço também a todos que direta ou indiretamente contribuíram para que fosse possível a realização deste trabalho, o meu muito obrigado!

“As tecnologias mais importantes são aquelas que desaparecem, elas se integram à vida do dia a dia ao nosso cotidiano até serem indistinguíveis dele” (Mark Weiser)

RESUMO

Neste trabalho é desenvolvido um dispositivo autônomo, de baixo custo e baseado em *IoT* com o propósito de monitorar as condições climáticas do ambiente de um *data center*. Para isso foram utilizadas a plataforma *Arduino*, um sensor de temperatura e umidade *DHT22*, um sensor de fumaça *MQ-2* e um módulo *Ethernet Shield* integrado a uma plataforma web, que permite ao usuário consultar o *status* da situação climática por meio de gráficos. Caso a temperatura e a umidade estejam em níveis críticos ou se for identificada a presença de fumaça, será possível receber alerta sonoro diretamente pelo dispositivo e por um aplicativo *Android*. Esse sistema visa identificar possíveis problemas ocasionados pela falta de infraestrutura necessária no *data center* ou defeitos ocasionais que levem a perigos iminentes. Assim, o presente trabalho busca garantir a alta taxa de disponibilidade dos sistemas cruciais para a organização, com a utilização de um sistema de monitoramento. Os resultados deste trabalho demonstraram um bom desempenho dentro da sua proposta, com um ótimo custo-benefício para solucionar o problema relatado. As implementações e melhorias em ambientes de missão crítica, como os *data centers*, não param de acontecer, porém, dada à relevância do tema e complexidade das tecnologias estudadas, faz-se necessário um aprofundamento constante sobre o assunto.

Palavras-chave: Internet das Coisas. Temperatura. Umidade. Data Center. Arduino.

ABSTRACT

This paper presents a stand-alone and low cost device for the purpose of monitoring the climatic conditions of the data center environment. The Arduino platform, a DHT22 temperature and humidity sensor, an MQ-2 smoke sensor and an Ethernet Shield module integrated to a web platform, allow the user to check the status of the weather through graphics. If temperature and humidity are at critical levels or if smoke is detected, you can receive audible alert directly through the device and through an Android application. This system aims to identify possible problems caused by the lack of necessary infrastructure in the data center or occasional defects that lead to imminent dangers. Thus the present work seeks to clarify that the adoption of a monitoring system has as main objective to guarantee high rate of availability of systems crucial to the organization. The results of this work demonstrated a good performance within its proposal, with a great cost-benefit to solve the reported problem. Implementations and improvements in mission-critical environments, such as data centers, do not stop happening, however, given the relevance of the theme and complexity of the technologies studied, it is necessary a constant deepening on the subject.

Keywords: *Internet Of Things*. *Temperature*. *Humidity*. Data Center. Arduino.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ANSI	American National Standards Institute
API	Application Programming Interface
APP	Applications
BTU	British Thermal Unit
CNC	Companhia Nacional de Cilindros
CO	Monóxido de Carbono
CPD	Centro de Processamento de Dados
GLP	Gás Liquefeito de Petróleo
GND	Graduated Neutral Density Filter
GND	Ground (Terra)
HTTP	Hyper Text Transport Protocol
HTTPS	Hiper Text Transfer Protocol Secure
I2C	Inter-Integrated Circuit
IdC	Internet das Coisas
IDE	Integrated Development Environment
IoT	Internet of Thing
IPV6	Internet Protocol Version 6
JSON	JavaScript Object Notation
LAN	Local Area Network
LCD	Liquid Crystal Display
LED	Light Emitting Diode
MAC	Media Access Control
MIT	Massachusetts Institute of Technology
MQTT	Mesage Queue Telemetry Trasnport
PWM	Pulse Width Modulation
RAM	Random Access Memory
RJ45	Registered Jack 45
SCL	Serial Clock
SD	Secure Digital
SDA	Serial Data
SMS	Short Message Service

SPI	Serial Peripheral Interface
TCP/IP	Transmission Control Protocol/Internet Protocol
TIA	Telecommunications Industries Association
URL	Uniform Resource Locator
USB	Universal Serial Bus
VCC	Voltage Common Collector
XML	EXtensible Markup Language

Sumário

INTRODUÇÃO

Há algum tempo, as tecnologias vêm crescendo de maneira significativa e têm ganhado lugar na vida cotidiana. Estamos em fase de transição com a denominada "Quarta Revolução Industrial", que poderá transformar o modelo econômico atual, principalmente, através da Internet das Coisas e *Big Data*, porém a transformação não ficará apenas no modelo econômico (O GLOBO, 2016)[1]. Segundo o portal GLOBO (2016)[2]:

> Estamos a bordo de uma revolução tecnológica que transformará fundamentalmente a forma como vivemos, trabalhamos e nos relacionamos. Em sua escala, alcance e complexidade, a transformação será diferente de qualquer coisa que o ser humano tenha experimentado antes (GLOBO, 2016)

Com a popularização da internet de banda larga fixa e móvel, houve um grande aumento no volume de informações e serviços disponíveis na internet, além do grande aumento do número de dispositivos conectados. Uma das tecnologias que vem mais se destacando é a *IoT*, ou Internet das Coisas, que tem como principal característica a conexão de objetos comuns com a internet, essa tecnologia consiste na ideia de unir os objetos físicos do "mundo real" com o "mundo digital", fazendo com que as pessoas possam se interagir com diversos objetos comuns do nosso cotidiano, dessa forma com a *IoT* é possível monitorar e gerir diversos ambientes ou equipamentos de forma inteligente e automática, diminuindo assim a mão de obra humana.

De acordo com o site *Business Insider*, é previsto que até 2020 existirão cerca de 24 bilhões de dispositivos de internet das coisas, enquanto os dispositivos tradicionais de computação (por exemplo, *smartphone, tablets, smartwatches* etc.) compreenderão cerca de 10 bilhões e serão gastos quase 6 trilhões de dólares em soluções *IoT* nos próximos 5 anos. (CAMHI, 2015)[3].

---

[1] O GLOBO. **Tecnologia pode acabar com 5 milhões de empregos no mundo até 2020**. Disponível em: https://oglobo.globo.com/economia/tecnologia-pode-acabar-com-5-milhoes-de-empregos-no-mundo-ate-2020-18498564. Acessado em: 27 de Jan de 2018.

[2] G1.GLOBO. **O que é a quarta revolução industrial e como ela pode afetar nossas vidas**. Disponível em: http://g1.globo.com/economia/negocios/noticia/2016/10/o-que-e-a-4a-revolucao-industrial-e-como-ela-deve-afetar-nossas-vidas.html. Acessado em; 27 de Jan de 2018.

[3] CAMHI, Jonathan, BI. **Intelligence projects 34 billion devices will be connected by 2020**. Disponível em: http://www.businessinsider.com/bi-intelligence-34-billion-connected-devices-2020-2015-11. Acesso em 08 de mar 2018.

Devido ao crescimento tecnológico atual, existe muitas possibilidades de aperfeiçoar a tecnologia, principalmente no que tange à segurança. Para isso surgem os chamados *data centers,* locais onde são instalados sistemas computacionais, de telecomunicações, de redes, armazenamento de dados das empresas e organizações. Tais equipamentos possuem um altíssimo valor aquisitivo, bem com informacional. Em uma sociedade tecnológica, o uso desses locais é indispensável e necessita de um ambiente adequado e extremamente seguro. Por esse motivo os *data centers* necessitam de um rigoroso padrão de construção e monitoramento contínuo.

A utilização dos d*ata centers* nas organizações aumentou significativamente nos últimos anos, principalmente após a popularização da Computação em Nuvem, pois houve um grande aumento de exigências de processamento e armazenamento de dados, com isso, o monitoramento desses ambientes se tornou algo essencial, pois uma falha nesses ambientes pode deixar todos os serviços indisponíveis. A falha pode ocorrer devido a parada de um dos equipamentos, por problemas de refrigeração ou energia elétrica, que as vezes podem causar danos irreparáveis.

As tecnologias e os conceitos para diminuir estes riscos estão evoluindo cada vez mais com o propósito de alcançar maior segurança e disponibilidade dos sistemas. Para isso a *Uptime Institute* desenvolveu o sistema de classificação de normas TIER, como um meio de mensurar o nível da infraestrutura de *data center* em termos de exigências de negócio para a disponibilidade e segurança dos sistemas.

Segundo a ANSI/TIA-942 (2005), a classificação dos *data centers* é composta pela avaliação individual de quatro áreas: telecomunicações, arquitetura, elétrica e mecânica, na qual é avaliada a classificação geral do *data center*. O sistema mecânico de um *Data center* é um dos mais importantes, principalmente quando se trata de refrigeração, pois uma das características dos equipamentos, principalmente servidores e nobreaks, é a grande dissipação de calor e a sensibilidade desses equipamentos à temperatura e umidade elevada, o que pode diminuir o tempo de vida e até mesmo a ocorrência de falhas, gerando grandes prejuízos para a organização.

Desta forma, o estudo se orientará através da seguinte problemática: Como prever e evitar problemas de segurança ocasionados pela alta temperatura, excesso umidade do ar e a presença de fumaça em virtude de falhas nos equipamentos do *data center* da Justiça Federal de Picos?

Devido à falta de recursos financeiros, em vista a crise econômica que assola o Brasil atualmente, na Justiça Federal de Picos não está sendo possível adquirir todos os componentes necessários para obter máxima segurança para o Data center, porém é possível diminuir consideravelmente os custos para solucionar esses problemas relacionados a falta de estrutura adequada com a utilização da IoT e a plataforma *Arduino*, com isso é possível criar um sistema para monitorar diversos sensores com o objetivo de prever e alertar diversos tipos de alterações no ambiente de um *Data center*.

A proposta deste trabalho é desenvolver um sistema autônomo, de baixo custo baseado em *IoT* para monitoramento e coleta de dados da temperatura, umidade relativa do ar e fumaça de um d*ata center*, como forma de controle e prevenção de danos aos equipamentos. O estudo de caso foi definido no município de Picos/PI, onde a coleta de dados se dará através da pesquisa de campo no *data center* Justiça Federal de Primeiro Grau, localizado na Rua Santo Antônio, 74 – Centro, CEP: 64600-004. Nesse local, o uso desta tecnologia permite o monitorar a situação climática em tempo real do citado ambiente.

Para o desenvolvimento deste projeto foi necessário: montar um protótipo com uma placa micro controladora *Arduino UNO,* um módulo *Ethernet Shield, um Display LCD 20x4,* Sensores *DHT22 e MQ-2 ;* desenvolver o software para o *Arduino UNO* coletar e enviar os dados dos sensores através do *Ethernet Shield*; criar e configurar uma conta na plataforma *Web Thingspeak* para receber os dados e alertar caso os dados estejam fora da faixa; e realizar testes do protótipo e do sistema a fim de garantir a coleta de dados necessárias para o sistema de forma eficaz.

Os *data centers* necessitam de um sistema de resfriamento efetivo devido à quantidade de equipamentos que geram calor e que devem ser controlados com centrais de ares-condicionados de precisão e sistemas de monitoramento ambiente. Também é necessário a utilização de um sistema de detecção de fumaça para aviso precoce, que identifica e alerta os usuários a fim de evitar danos e perdas durante um princípio de incêndio (ANSI/TIA-942, 2005).

Segundo Veras (2009), a ANSI/TIA-942 é a norma internacional que especifica os requisitos mínimos para infraestrutura de *Data center*, de acordo com o grau de disponibilidade e redundância desse sistema, a fim de garantir a integridade dos equipamentos eletrônicos.

Nesta monografia a pesquisa é direcionada na área mecânica, que é composta por sistemas de climatização e de proteção contra incêndio. De acordo com Veras (2009), os sistemas de refrigeração devem ser capazes de atingir as temperaturas de 20ºC a 25ºC, e a umidade relativa do ar deve ser controlada entre 45% e 55%. A grande dificuldade é manter o ambiente nessas condições, visto que o seu funcionamento é ininterrupto, para isso é necessário a utilização de ar condicionado de precisão, que são equipamentos específicos para esse tipo de ambiente.

No mercado já existem diversos tipos de equipamentos específicos que podem suprir a necessidade de monitoramento de *data center*, porém o custo é bastante elevado pois existe pouca concorrência e normalmente poucos componentes são fabricados no Brasil. Além disso, esses equipamentos são pouco flexíveis pois são sistemas proprietários.

O alto custo desses equipamentos e a falta de recursos do governo afetam bastante a evolução desse processo. É o que ocorre na Justiça Federal de Picos-PI, pois utilizam-se ar condicionados comuns e ineficientes, o que leva ao risco de falhas, como desligamentos inesperados e vazamentos de água por falta de manutenção. Na parte elétrica também há um risco de algum componente do d*ata center* queimar, e o fogo se alastrar causando, assim, danos nos equipamentos. O impacto pode ser grande, e os funcionários podem não perceber, pois a sala é bem isolada.

No local de pesquisa, os dois ares-condicionados são ligados alternadamente em período de doze horas, visando-lhes minimizar o desgaste, que já são bem antigos. Com isso fica evidente uma situação precária que justifica a necessidade de utilizar um sistema de monitoramento para ajudar a identificar possíveis problemas que possam surgir relacionados à temperatura inadequada ocasionada por alguma falha na refrigeração, além do risco de causar oxidação de componentes eletrônicos por excesso de umidade.

A utilização de um sistema com *Arduino* e a IoT trará uma redução significante nos custos para a implantação, visto que um equipamento para atender a essa demanda custa em média R$ 2.990,00 [3]; em contrapartida, o *kit Arduino* com todos os materiais necessários custará em torno de R$ 185,10 [4], ou seja, em torno de 93,8% mais barato. É interessante observar que o protótipo *Arduino* tem algumas funcionalidades diferenciadas, pois ele, além de monitorar temperatura e umidade

relativa, também monitora a fumaça no ambiente e tudo pode ser visualizado pela internet, além de emitir alarme sonoro e visualização das informações em gráficos.

O *Arduino* se destaca por sua facilidade para desenvolver projetos de automação e de internet das coisas, pois possui uma vasta documentação na internet, fóruns e cursos gratuitos, desta forma, pessoas com conhecimento básico em eletrônica e programação conseguem facilmente desenvolver projetos com essa plataforma. Neste projeto o código pode ser alterado e outros sensores também podem ser facilmente acrescentados, por exemplo, um sensor de corrente para medir o consumo de energia elétrica ou um sensor de calor para identificar a presença de fogo. Desta forma o *Arduino*, por ser uma plataforma de código aberta, é uma ferramenta flexível e eficiente na solução do problema proposto e com um baixo custo.

---

[3] Preços obtidos em www.prodigital.com.br

[4] Preços obtidos em www.piauino.com.br

Os procedimentos metodológicos para o desenvolvimento do estudo são de natureza bibliográfica, exploratória e de campo, uma vez que busca compreender a sua aplicação na vida real das organizações. Para a elaboração desta monografia, a metodologia utilizada foi a pesquisa definida, assim, por Rampazzo (2005), como procedimento reflexivo, sistemático, controlado e crítico, que permite encontrar novas soluções, fatos, dados ou regras, em qualquer área do saber. Assim, as pesquisas bibliográficas foram elaboradas a partir de revistas, artigos acadêmicos, bibliografias diversas, dentre outros. Para o desenvolvimento do um protótipo, foi necessário:

1) Desenhar o projeto utilizando o software *Fritzing*;
2) Montar projeto com *Arduino*, *Shield Ethernet*, Sensores, *Display*, *Buzzer* e um Led;
3) Desenvolver o código utilizando o *Arduino IDE* para coletar os dados dos sensores;
4) Implementar a comunicação com o servidor *Thingspeak* para envio dos dados;
5) Instalar e configurar o aplicativo mobile do *Thingspeak* para melhor visualização em smartphones.

Este sistema irá analisar os dados obtidos e enviará alerta por mensagens caso sejam identificados temperatura e umidade elevadas ou a presença de fumaça. Em seguida, será realizado um teste em um ambiente simulado por 2 dias, depois o protótipo será instalado no *Data center* para realizar os testes em ambiente real

durante 2 dias. Após os testes, serão analisados os resultados obtidos por meio de gráficos a fim de verificar se houve falhas nas aquisições de dados. Para isso, é necessário verificar se em algum momento a leitura obteve o valor zero ou nulo, que podem ser facilmente visualizados por gráficos. Além disso, serão apresentados comparativos por meio de figuras com o objetivo de auxiliar a análise proposta do tema discutido.

Estima-se que, para o desenvolvimento do sistema, será necessário um investimento de aproximadamente R$ 185,10. Os valores da tabela a seguir informam apenas o custo dos equipamentos comprados, não foram calculados os custos de frete e mão de obra. (Tabela 01).

Tabela 01 – Estimativa de Custo (PIAUINO, 2018)

QUANTIDADE	DESCRIÇÃO	MODELO	CUSTO UNITÁRIO	CUSTO TOTAL
1	PLACA *ARDUINO*	UNO REV3	35,00	35,00
1	*ETHERNET SHIELD*	R3	49,90	49,90
1	DISPLAY LCD C/ I2C	20X4	37,90	37,90
1	ALARME BUZZER CONTÍNUO	5VCD	2,50	2,50
3	LEDS	5MM	0,30	0,90
1	SENSOR DE TEMPERATURA	AM2302 *DHT22*	22,00	22,00
1	SENSOR DE GAS/FUMAÇA	*MQ-2*	17,90	17,90
1	PROTOBOARD	400 PINOS	14,00	14,00
1	CABO DE REDE	RJ-45	5,00	5,00
INVESTIMENTO TOTAL				**R$: 185,10**

Fonte: Autor (2018)

2 REFERENCIAL TEÓRICO

Para Lakatos e Marconi (2010, p. 224): “É imprescindível correlacionar a pesquisa com o universo teórico, optando-se por um modelo teórico que serve de embasamento à interpretação do significado dos dados e fatos colhidos ou levantados”.

A discussão sobre Internet das coisas e *Data centers* é antes de tudo, um pouco desafiante, pela a complexidade que o tema revela, que requer um maior estudo dessas tecnologias que evoluem rapidamente. Neste trabalho, apresentam-se pesquisas bibliográficas que servirão de embasamento teórico para o desenvolvimento do sistema proposto, na qual serão abordados os conceitos de Data center e as suas características, como por exemplo, sua segurança física, componentes de infraestrutura, refrigeração, normas técnicas e suas classificações, além de abordagens sobre *Middleware*, Internet das Coisas, plataforma *Arduino* e seus componentes, tudo dentro da perspectiva de implantação de um sistema de monitoramento climático para *Data Center*.

2.1 *DATA CENTER* OU CPD

Data center é um local onde são armazenados todos os dados e informações de uma empresa. Ele é construído para abrigar diversos servidores, com bancos de armazenamento de dados, os quais processam e mantém uma enorme quantidade de informação. Sua estrutura é composta por um piso elevado, com espaço para a passagem de cabos abaixo e uma sala climatizada (com ar-condicionado), para resfriar os computadores, que deve ser totalmente protegida contra incêndio e qualquer outra eventualidade que comprometa a integridade e segurança dos dados, (ADAMI, 2018).

O *Data center* possui um papel fundamental para o funcionamento dos diversos serviços da nossa rotina. É um ambiente especial onde são centralizados servidores, equipamentos de processamento e armazenamento de dados, *switches* de redes, roteadores, *Nobreak* e Estabilizadores, com o objetivo de armazenar milhões de servidores de banco de dados, estes equipamentos geralmente são montados em racks ou armários metálicos (SOBRAL, 2015). De acordo Driemeyer (2016), conforme

citado por Marin (2013), *Data centers* são ambientes de missão crítica que abrigam equipamentos responsáveis pelo processamento e armazenamento de informações nos mais variados tipos de organizações, como por exemplo, empresas privadas, órgãos públicos, escolas, hospitais, entre outros.

Segundo Driemeyer (2016), conforme citado por Filho (2016), Os *Data centers* vem se tornando uma tecnologia essencial no aperfeiçoamento dos sistemas computacionais e da internet. Devido a uma rápida evolução nesses sistemas, os conceitos de Data center mudaram bastante, e hoje são avaliados por normas para que haja maior disponibilidade e segurança dos serviços.

Nesse sentido, é possível perceber que a segurança do Data Center é muito importante, os mesmos são projetados para serem extremamente seguros e equipados com sistemas de monitoramento para evitar incêndios e danos aos equipamentos. O acesso às dependências do data center é feito com leitor biométrico ou cartões eletrônicos. (Figura 01).

Figura 1 - *Data Center*

Fonte: http://cwbmetal.com.br/wp-content/uploads/2015/01/como-e-um-data-center.jpg

Os *Data centers* possuem uma infraestrutura bem complexa e dinâmica na qual podemos citar segundo Fey (2014) que o sistema de cabeamento estruturado é formado por elementos padronizados denominados de subsistemas onde sua principal função é padronizar as instalações e se adaptar as alterações de *layout* da instituição, sem a necessidade de novas instalações de cabeamento.

A climatização do *Data center* também é muito importante, por isso deve-se ficar atento a alguns fatores para sua implantação, como: o tipo de ar condicionado adequado ao tipo de *Data center*, o direcionamento do fluxo de ar; altura entre o piso e o teto da sala, pois cada um tem seus requisitos de climatização.

Marin (2011, p. 15) explica que um *Data center* é classificado por três atributos essenciais:

> Disponibilidade: é o tempo durante o qual o sistema está em funcionamento onde deve ser no mínimo, 99,67% e para isso é essencial um sistema redundante capaz de suprir a parada de seus sistemas principais;
> Confiabilidade: a alta confiabilidade de um sistema pode ser entendida que ele contenha muitos componentes confiáveis, pois é definida como possibilidade de não ocorrência de falhas antes de uma quantidade de horas;
> Redundância: tem a finalidade de diminuir e/ou evitar o tempo de parada de um sistema. Para isso, devem-se duplicar os componentes e sistemas para prevenção e recuperação de sinistros como erros, técnicas, falhas humanas e manutenção do sistema.

No que se refere à parte elétrica, Marin (2011) informa que o passo inicial na execução da planta elétrica é definir a classificação TIER a ser utilizada e conhecimento da carga a ser requerida. Pois, se for subestimada, pode comprometer as operações devido às expansões. Por outro lado, se forem superestimadas, haverá desperdício de dinheiro.

2.1.1 Normas Para a Classificação do *Data Center*

Existem diversas normas para mensurar o nível de qualidade e segurança de *Data centers*, porém, a mais utilizada é a norma TIER, que Segundo Marin (2011, p. 17), explica que essa norma foi desenvolvida em 2009 pela *The Uptime Institute*, que é uma instituição de certificação de *Data centers* que lançou a norma de classificação TIERS para infraestrutura com base nos conceitos de redundância de e tolerância a falhas. O Instituto elaborou quatro classificações TIER de infraestrutura para centro de dados:

> *Data center* tier I: Este é o modelo básico, pois não possui componentes redundantes em seu sistema elétrico ou de refrigeração. Tem apenas a configuração mínima e, portanto, devem ser desligados para manutenção;
> *Data center* tier II: Neste centro de dados existem componentes redundantes, porém somente uma rede de distribuição elétrica e de refrigeração para atender a carga crítica de TI. Portanto, devem ser desligados para manutenção preventiva e corretiva;
> *Data center* tier III: Este modelo pode ocorrer manutenção e operação simultâneas, pois existem componentes e encaminhamentos redundantes para atender a carga crítica de TI em que todos os equipamentos serão instalados com dupla fonte de alimentação de energia. Assim, as atividades de manutenção preventivas podem ser realizadas.

> *Data center* tier IV: É um modelo tolerante a falhas, pois seus sistemas, componentes, encaminhamentos, sistema elétrico e de refrigeração são independentes e dupla fonte de alimentação. Mantém a capacidade de manutenção preventiva ou corretiva e não são vulneráveis a falhas por abalos isolados

A tabela 02 a seguir mostra de forma mais resumida os itens que são exigidos em cada nível da norma TIER da classificação mecânica:

Tabela 02 – Classificação Mecânica da TIER

MECÂNICA	**TIER 1**	**TIER 2**	**TIER 3**	**TIER 4**
Tubulação próximo ao datacenter	Permitido mas não recomendado	Permitido mais não recomendado	Não permitido	Não permitido
Sistemas de climatização ligados aos gerador principal e reserva	Sem exigência	Sim	Sim	Sim
Unidade de ar condicionado interno	Sem redundância	Uma unidade redundante por área crítica	Unidades de ar condicionado, suficientes para manter a área crítica durante uma perda de uma fonte de energia.	Unidades de ar condicionados, suficiente para manter a área crítica durante a perda de um fonte de energia elétrica
Controle de umidade relativa do ar	Sim	Sim	Sim	Sim
Fonte de energia elétrica redundantes para o sistema de climatização	Não	Não	Sim	Sim
Sistema de detecção de incêndios	Não	Sim	Sim	Sim
Sistema de extintores de incêncio	Quando solicitado	Pré-ação(quando necessário)	Pré-ação(quando necessário)	Pré-ação(quando necessário)
Sistema de supressão gasosa	Não	Não	Agentes limpos listados na NFPA 2001	Agente limpos listados na NFPA 2001
Sistema de Detecção de Fumaça de Aviso Precoce	Não	Sim	Sim	Sim
Sistema de detecção de vazamentos de água	Não	Sim	Sim	Sim

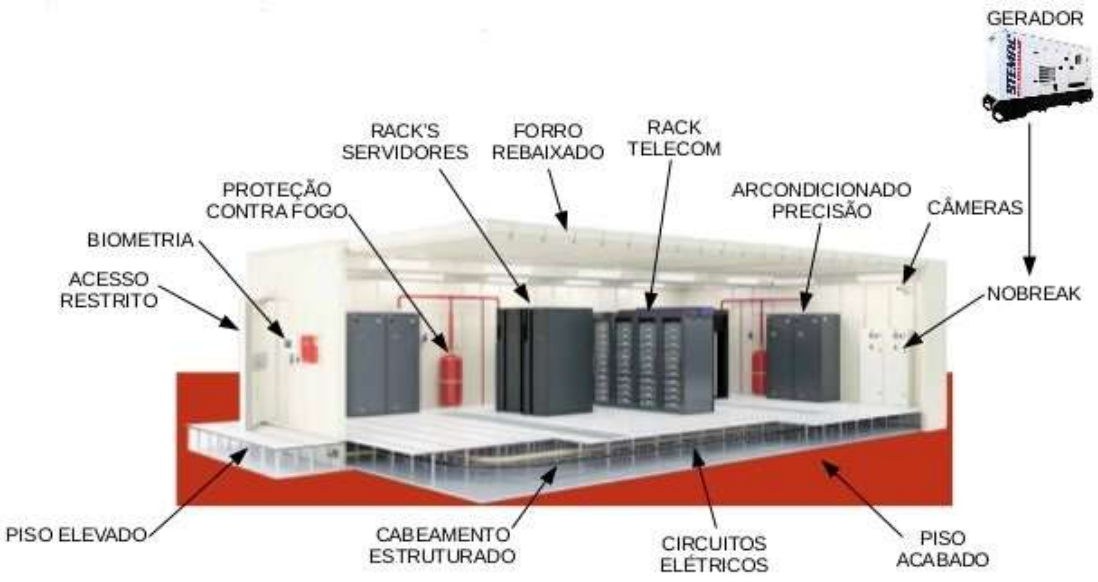
Fonte: Adaptado de ANSI/TIA-942 (2018)

A norma TIER I da classificação Mecânica inclui unidade de ar condicionado comum com potência suficiente para manter a temperatura e umidade relativa ideal no espaço, porém sem unidades redundantes, que pode causar interrupção em caso de manutenções. Se o *Data center* possuir gerador, todos os equipamentos de ar condicionado devem ser alimentados por ele (ANSI/TIA-42, 2005).

No TIER II, além das características do TIER I, o sistema de climatização deve possuir várias unidades de ar condicionado com a capacidade combinada de resfriamento, para funcionar de forma redundante, além disso deve o sistema deve ter capacidade para operar de forma contínua, durante 365 dias / ano (ANSI/TIA-42, 2005). De acordo com Marin (2011), no TIER III deve possuir os requisitos do TIER II, além de unidades redundantes suficiente para minimizar falhas ou manutenções no quadro elétrico. Este nível de redundância pode ser adquirido quando é fornecida duas fontes de energia para cada ar condicionado (ANSI/TIA-942). Para o TIER IV, além de cumprir os requisitos do TIER III, há necessidade de redundância 2(N+1) conforme ANSI/TIA-942 (2005).

Um *Data center* requer cuidados básicos preventivos, corretivos e adequações para garantir uma alta disponibilidade na organização, aumentando assim também a vida útil dos equipamentos. A figura 2 mostra os componentes básicos de um *Data center*:

Figura 2 - Infraestrutura Básica Do *Data Center*



Fonte: https://pt.slideshare.net/LucianoVargas1/data-center-instituto-federal-farroupilha

2.1.2 Ar Condicionado *Split* X Ar Condicionado De Precisão

Neste tópico será descrito sobre os dois principais tipos de sistemas de climatização em relação à classificação mecânica de *Data center*.

O ar-condicionado *Split* ou de “conforto” (Figura 3), é o tipo de ar condicionado mais utilizado nas organizações, e foi desenvolvido para operar em calor latente ou úmido, ou seja calor gerado por seres vivos, animais e plantas, e é utilizado para manter o ambiente confortável em estabelecimentos com fluxo de pessoas, este tipo de ar-condicionado consome bastante energia elétrica, pois além de refrigerar também retira a umidade do ambiente, por isso este tipo de equipamento tem um trabalho mais “pesado”, e foi desenvolvido para suportar um uso diário de até 10 horas. O ar-condicionado *Split* não é recomendado para uso em *Data centers*, pois o mesmo além de consumir mais energia, não suporta um funcionamento ininterrupto 24 horas durante os 365 dias do ano.

Figura 3 - Ar-Condicionado *Split*

Fonte**:** https://www.magazineluiza.com.br/ar-condicionado-split-springer-midea-12000-btus-frio-38kcx12s5/p/0114480/ar/arsp/

O ar condicionado de precisão funciona de forma semelhante aos de *Split*, porém são bem mais robustos, pois são fabricados exclusivamente para utilização em ambientes de missão crítica e são projetados para operar em calor sensível ou latente, ou seja, é o calor gerado por equipamentos eletrônicos, que não geram umidade, pois raramente há pessoas circulando dentro dessas salas. Apesar do custo bem mais elevado, o ar condicionado de precisão gera uma grande economia de energia e redução de custos com manutenção. (Figura 4).

Figura 4 - Ar- condicionado de Precisão

Fonte: http://rmsenergia.com.br/produtos/sistema-de-ar-condicionado-de-precisao/

2.2 A INTERNET DAS COISAS OU *IOT*

Segundo Oliveira et. al (2016, p. 31), a Internet das Coisas ou *IoT* é:

> [...] uma grande rede de dispositivos interconectados encontrados no cotidiano, desde máquinas industriais até bens de consumo, que podem compartilhar informações e executar tarefas através de comandos de software. Os dispositivos são conectados de forma inteligente, através de comandos realizados por programação.

Acredita-se que a discussão sobre a conexão de objetos começou no ano de 1991, quando a conexão de TCP/IP (*Transmission Control Protocol/Internet Protocol*) e a Internet que é conhecida atualmente foram se desenvolvendo para tornar-se acessíveis.

A plataforma *IoT* normalmente possui uma interface com o usuário através de uma aplicação web e conversa com os seus dispositivos micro controlados – *Arduino*, *Nodemcu*, *Raspberry* e outros -, armazenando ou controlando os dados monitorados. Segundo Zaslavsky et al. (2013, p. 03), "a Internet das Coisas tem o potencial de mudar o mundo, assim como a Internet fez, talvez até mais". Para Atzori et al. (2010), a *IoT* vem ganhando grande destaque no cenário das telecomunicações e está sendo considerada a revolução tecnológica que representa o futuro da computação e comunicação.

Um dos motivos da Internet das Coisas estarem se tornando cada vez mais disponíveis se deve ao fato dos preços de hardware estarem constantemente diminuindo. Projetos como *RaspberryPI* e *Arduino* fornecem componentes de hardware modulares de baixo custo que podem ser facilmente programados e configurados.

De acordo com Freitas (2016), a Internet das coisas é um grande desafio, tanto em seu nível conceitual, como também tecnológico, pois são várias tecnologias embutidas num sistema. Há muitos segmentos de mercado e verticais com aplicação de internet das coisas e cada vez mais surgem novas aplicações.

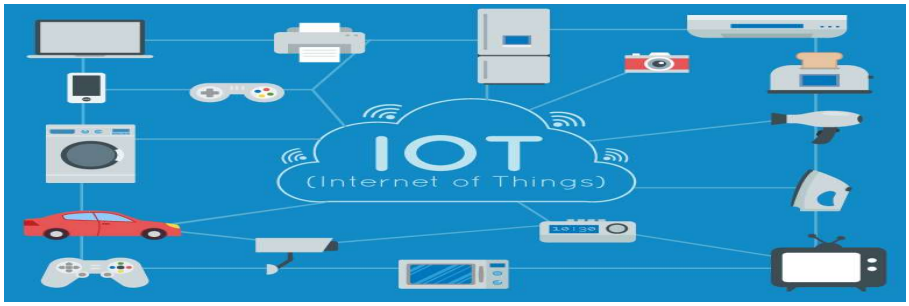
De acordo com a Manyika (2015, p. 15):

> [...] o impacto econômico da internet das coisas será de 3,9 a 11,1 trilhões por ano em 2025, significando 11% da economia mundial. Neste cenário, os usuários serão o maior potencial econômico, cerca de US$ 7,5 trilhões, que lhes trará maior conveniência, melhores produtos e serviços com o uso de sistemas de Internet das Coisas para capturar esse valor. Por outro lado, abre uma gama de oportunidades de projetos às empresas. No mundo da Internet das Coisas, a tecnologia digital é parte da estratégia empresarial e serão necessários novos modelos de gestão empresarial e de negócios, que vejam a tecnologia como fonte de vantagem competitiva e não apenas como função de suporte

Outra questão altamente relevante é a segurança para *IoT*, este ponto, por ser uma tecnologia ainda relativamente nova, é uma das principais barreiras que impedem a efetiva adoção da *IoT* em massa, pois os usuários ainda estão preocupados com a violação dos seus dados. Deste modo, a segurança desempenha papel chave para a adoção da *IoT*.

De forma prática podemos entender internet das coisas como objetos comuns do nosso cotidiano que são integrados à dispositivos eletrônicos que tem a capacidade de coletar dados dos objetos e compartilhar com a internet ou até mesmo ser controlado remotamente, na Figura 5, é possível visualizar uma demonstração de diversos tipos de objetos que são conectados à internet.

Figura 5 - Demonstração de objetos conectados à Internet



Fonte: https://becode.com.br/internet-das-coisas/

2.3 MIDDLEWARE

De acordo com Cunha (2006), *Middleware* é um sistema mediador capaz de fornecer uma abstração do sistema para as aplicações e para os desenvolvedores de aplicações, abstraindo a complexidade dos mecanismos de hardware, software e interfaces de comunicação. Dessa forma, a padronização de uma camada de middleware permite a construção de aplicações independentes do hardware e do sistema operacional, executáveis em qualquer plataforma de qualquer fabricante, assim o principal objetivo do *Middleware* é intermediar a interação de aplicações com as fontes de dados.

De acordo com empresa consultora de mercado MarketsandMarkets, o mercado de *middleware* para Internet das Coisas (*IoT*) crescerá de US$ 3 bilhões em 2015 para US$ 11 bilhões em 2020. As regiões da América do Norte e Europa deverão impulsionar esses valores, mas o crescimento será sentido em todas as regiões do globo (MARKETSANDMARKETS, 2015).

Com a utilização de um *Middleware*, o desenvolvimento de uma aplicação *IoT* se torna muito mais fácil, pois ao desenvolver um sistema embarcado para enviar dados para internet, o *Middleware* se encarrega de receber, tratar e armazenar esses dados, realizando diversas tarefas bastante interessantes, como: geração de relatórios, visualização de gráficos em tempo real, envio de mensagens de alerta por *SMS*, e-*mail* ou redes sociais com a utilização de diversas API que vem junto com o *Middleware* ou até mesmo reaproveitar esses dados gerados em outro sistema *web* particular.

2.3.1 IOT ANALYTICS THINGSPEAK

O *ThingSpeak* é uma plataforma *IoT* do tipo middleware que permite criar diversos canais para analisar, visualizar e armazenar os dados em nuvens. O *Thingspeak* pode receber dados enviados de dispositivos como *Raspberry* e *Arduino*, por exemplo. Dessa forma é possível configurar os dispositivos para coletar e enviar dados para a plataforma *Thingspeak*, que irá receber e mostrar os dados em campos com gráficos personalizados. Também estão inclusos a possibilidade de enviar alertas pré-definidos para dispositivos usando serviços *Web* como *Twitter* ou *Twillio*.

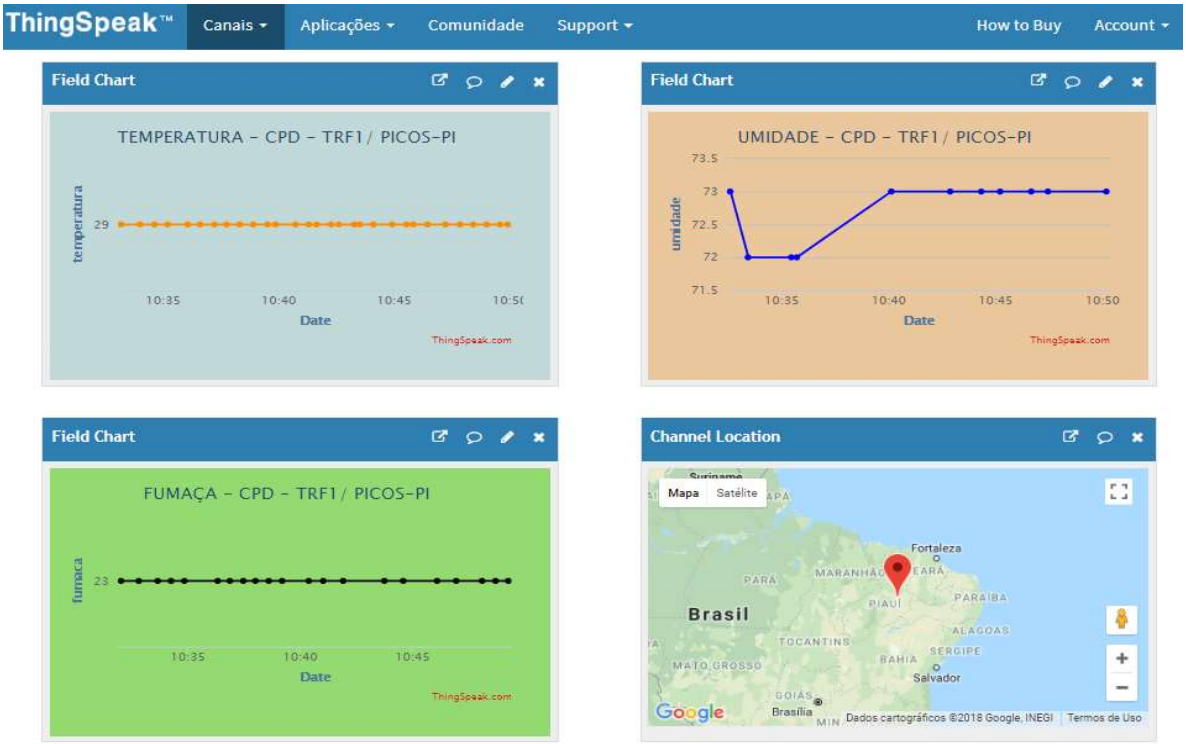
Esses recursos são bastante interessantes para serem usados nos setores de monitoramento de grandes organizações, pois é possível criar diversos canais para monitorar sensores em qualquer local com conexão à internet, por exemplo pode-se

criar um canal para monitoramento de um *Data center* e outro para monitoramento de uma câmara frigorífica, ou monitorar uma rede de lojas. Dentre outros recursos disponíveis, é possível gerar relatórios dos dados em XML, JSON e SVC, esses relatórios servem para realizar análises mais detalhadas ou importar para outros bancos de dados, para outras utilidades.

O *Thingspeak* demonstra ser uma ferramenta segura pois permite que usuário configure os canais como públicos ou privados, por padrão os canais são privados e requerem uma chave API para visualizar as informações, caso o usuário opte por um canal público, qualquer usuário pode acessar as informações do canal através de um endereço *Web.* É importante observar alguns detalhes, que na conta gratuita do *Thingspeak* só é possível receber dados enviados pelos dispositivos ou software a cada 15 segundos.

O *Thingspeak* é uma ferramenta bastante interessante, pois é possível criar sistemas *IoT* gratuitamente sem configurar servidores ou desenvolver sistemas Web e se destaca pelas diversas ferramentas de tratamento de dados e por sua facilidade de uso, tendo diversos artigos e fóruns na internet explicando como utilizar. Na figura 6, mostra exemplos de gráficos com os dados monitorados.

Figura 6 – *Thingspeak*



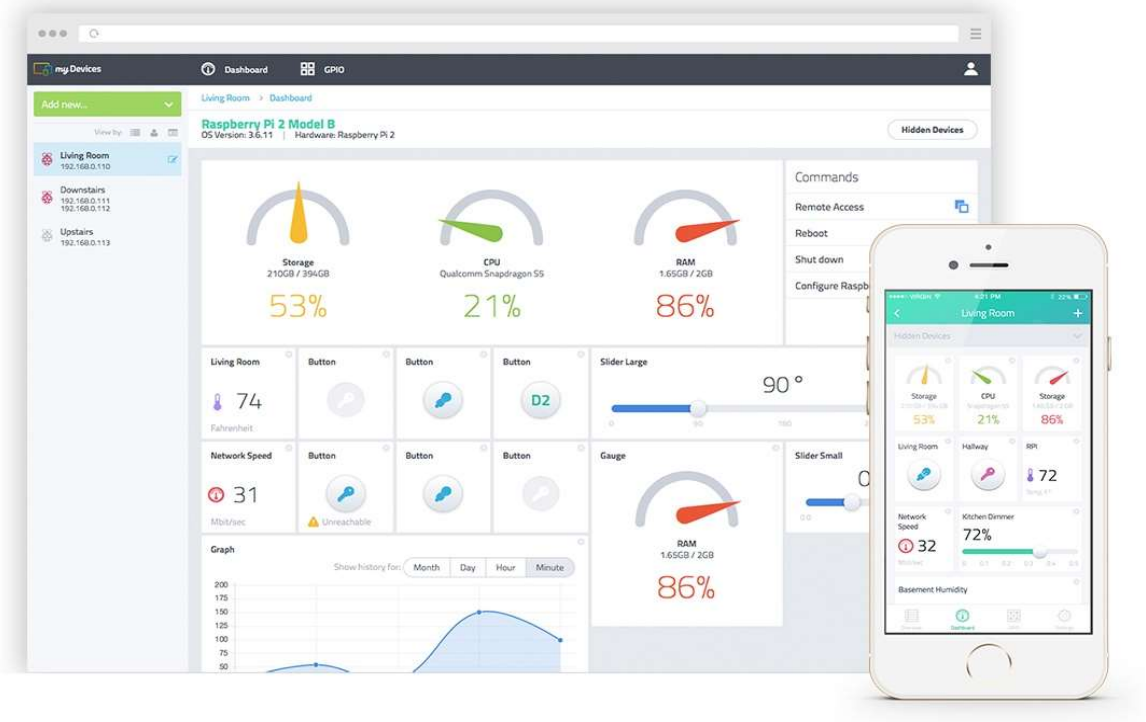
Fonte: autor, canal disponível em: https://thingspeak.com/channels/36813

2.3.2 CAYENNE MYDEVICES

É uma plataforma *IoT* do tipo *Middleware* que permite desenvolver rapidamente um projeto *IoT*, integrar ao protótipo e comercializar soluções, nele é possível criar soluções como: edifícios inteligentes, agricultura inteligente, medições inteligentes e aplicativos de rastreamento de ativos. O *Cayenne* permite que fabricantes de sensores, gateways, micro controladores e outros dispositivos façam com que seus equipamentos ofereçam a capacidade de "arrastar e soltar" para a sua plataforma, isso significa que um desenvolvedor de *IoT* pode integrar em seu projeto um dispositivo habilitado para *IoT* sem precisar escrever o código para abastecer esse mesmo dispositivo. O objetivo é tornar mais fácil e rápido o desenvolvimento de uma solução de *IoT*.

No *Cayenne* é possível criar um número ilimitado de projetos no serviço da nuvem *Cayenne* e também há um armazenamento ilimitado para dados de sensores, disparadores configuráveis e alertas, é possível criar painéis com *drag-drop* e *widgets* personalizados facilmente. Pelo *Cayenne* também é possível visualizar os dados de forma bem detalhada através do *Dashboard*. (Figura 7).

Figura 7 - *Dashboard* do *Cayenne*



Fonte: https://*IoT*.do/mydevices-cayenne-worlds-first-*IoT*-project-builder-2016-01

Em relação aos *middlewares*, os dois são gratuitos e fácil de utilização, eles armazenam dados em nuvens e fazem integração com ferramentas externas para fazer análise destes dados.

2.4 PLATAFORMA *ARDUINO*

O *Arduino* é um kit de desenvolvimento de prototipagem eletrônica baseada em *hardware* e *software* open source, são flexíveis e fáceis de utilizar como demarca Mcroberts (2011), onde são chamados de *open source* aqueles em que qualquer pessoa pode modificar e distribuir, tornando-o desta forma, totalmente acessível ao público.

Mcroberts (2011), afirma que a maior vantagem do *Arduino* sobre outras plataformas é facilidade de uso, pois qualquer pessoa pode facilmente aprender a criar seus próprios projetos em pouco tempo, e sem conhecimento especializado em eletrônica graças a sua linguagem de programação de fácil aprendizagem e por não precisar de conhecimento avançado em eletrônica, pois não é necessário soldar os componentes durante o desenvolvimento de projetos.

Em termos técnicos o *Arduino* é uma placa equipada com um microcontrolador *Atmel* e circuitos de entradas e saídas que pode ser conectado à um computador e programado através de um ambiente de desenvolvimento, utilizando a linguagem C/C++, baseada em *Wiring*, linguagem essa, de fácil aprendizagem.

Após realizar a programação do *Arduino*, ele pode ser usado de forma independente de conexão à computadores, ou seja, você pode colocá-lo para controlar e monitorar sensores, atuadores e relés. Dessa forma você pode automatizar tarefas, controlar robôs, luzes, temperatura do ar condicionado ou qualquer tipo de ideia que vier na sua cabeça, sendo que quase tudo pode ser monitorado ou controlado pela internet utilizando uma conexão de rede e um sistema web (MCROBERTS, 2011).

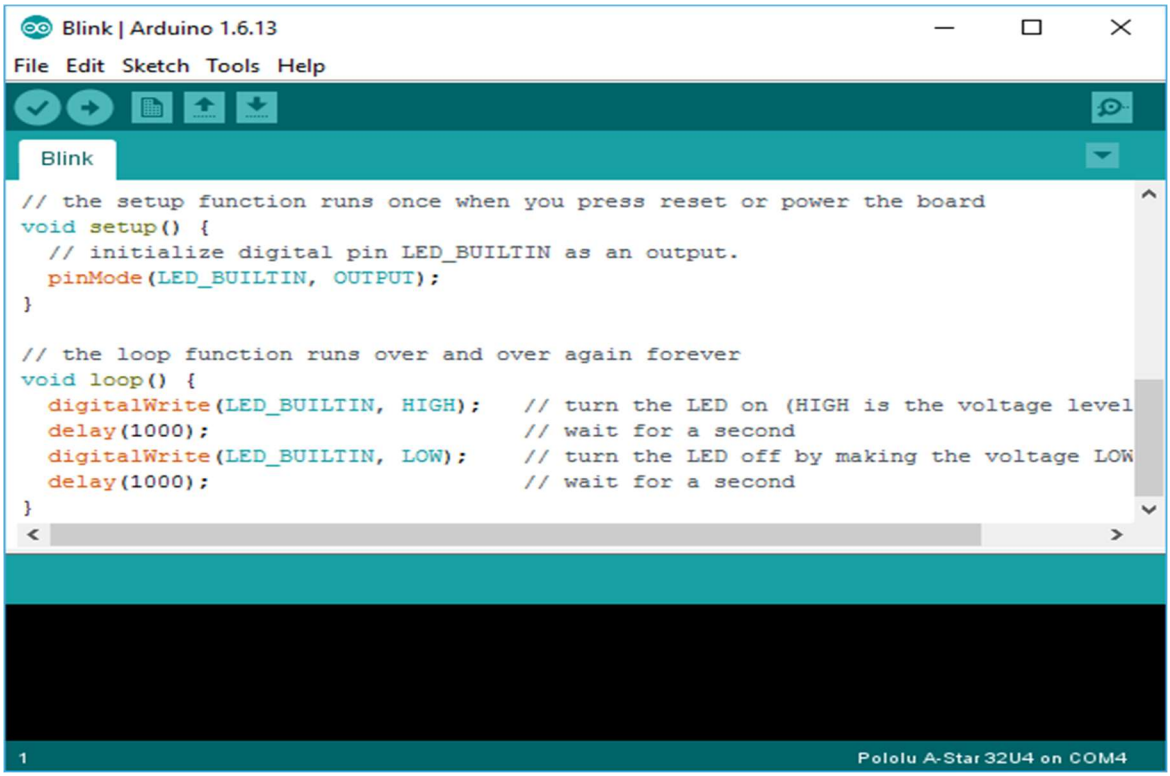
Para Frizzarin (2016, p. 12) “é possível pensar em Arduino pela ótica da Internet das Coisas (ou *Internet of Things - IoT*) que visa criar equipamentos com capacidade e objetivo de transmitir dados e interagir diretamente com a internet sem a efetiva intervenção humana”.

2.4.1 *ARDUINO* IDE

O *Arduino* IDE (Integrated Development Environment ou Ambiente de Desenvolvimento Integrado), é um software desenvolvido em Java para o *Arduino*, que contém um editor de texto para escrever o software, com uma série de menus e uma barra com botões para funções mais comuns que ajuda a agilizar o processo de codificação, pois nele é possível compilar o código, identificar e corrigir defeitos no código-fonte e visualizar o programa compilado em um monitor serial, há também um gerenciador de bibliotecas que ajuda a instalar e verificar as bibliotecas instaladas, além disso também vem uma variedade de exemplos de códigos fontes básicos para facilitar o aprendizado.

Nesta IDE deve ser inserida as funções *setup*() e *loop*(), e elas devem ser executadas mesmos que os escopos estejam vazios, pois é necessário para que o programa funcione. Na função *setup*(), devem ser inserida as instruções que só precisam ser iniciada uma vez, ou seja ela só será inicializada apenas uma vez quando a placa é ligada ou reiniciada, já na função *loop*(), todas a instruções que forem inseridas será sempre executada até que o *Arduino* seja desligado da fonte de alimentação. (Figura 8).

Figura 8: Arduino IDE



Fonte: https://www.pololu.com/docs/0J61/6.2

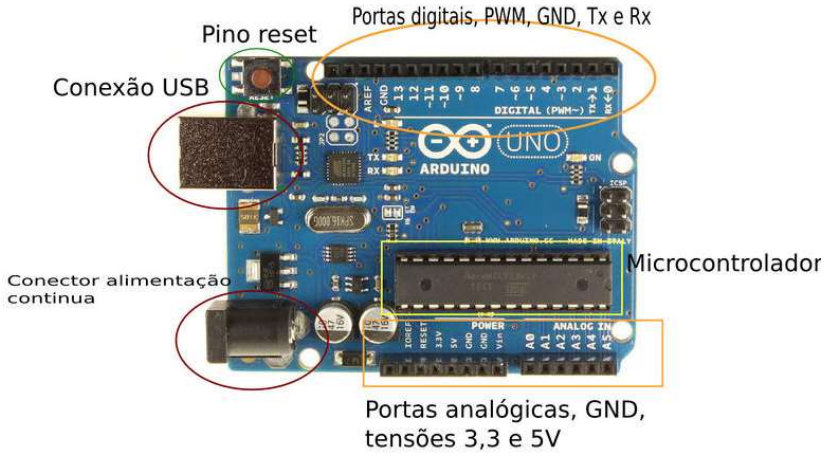
2.4.2 *ARDUINO UNO*

Este é um dos mais simples devido à sua baixa capacidade de armazenamento e processamento, porém é o mais utilizado de todos. De acordo com Thonsen (2014), é ideal para quem está iniciando na área de eletrônica e programação. Além disso, é um modelo mais econômico e compatível com a maioria dos *Shields* disponíveis para venda no mercado. (Figura 9).

As especificações básicas do *hardware* do *Arduino* são:

- Micro controlador *ATmega328*
- Voltagem Operacional de 5V
- 14 pinos digitais *I/O*
- 6 pinos analógicos de entrada
- Memória *Flash* de 32KB (*ATmega328*) dos quais 0,5 KB são utilizados pelo *bootloader*

Figura 9 - *Arduino* Uno



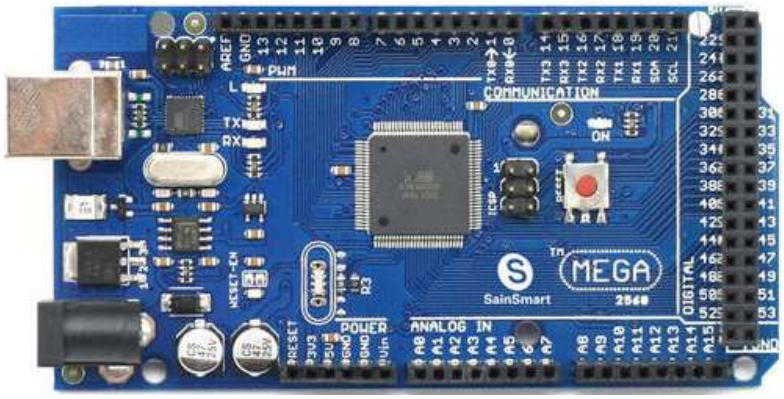
Fonte: https://wiki.sj.ifsc.edu.br/wiki/index.php/*Arduino*_-_Vis%C3%A3o_Geral

2.4.3 *ARDUINO MEGA*

O *Arduino* Mega (Figura 10), de acordo com Mukhopadhyay (2014), possui uma quantidade maior de portas, são 54 portas digitais e 16 portas analógicas, e maior capacidade de processamento e armazenamento em relação ao modelo *Uno*. Este, por sua vez é usado em projetos maiores que demandam maior capacidade do

dispositivo. De acordo com Vieira (2013), o *Mega* foi o primeiro *Arduino* com processador ARM 32 bits. Utilizado em projetos de larga escala.

Figura 10 - *Arduino Mega*



Fonte: https://store.*Arduino*.cc/*Arduino*-mega-2560-rev3

2.4.4 *ARDUINO YÚN*

Segundo Ribeiro (2014), *Arduino Yún* (Figura 11), é placa projetada para Internet das Coisas, por isso utiliza o nome Yún que significa “Nuvem” em chinês, esta placa visa a conectividade entre dispositivos, pois o mesmo já vem com uma entrada de rede *RJ45* integrada e um módulo *WiFi*.

Figura 11 - *Arduino* Yún.

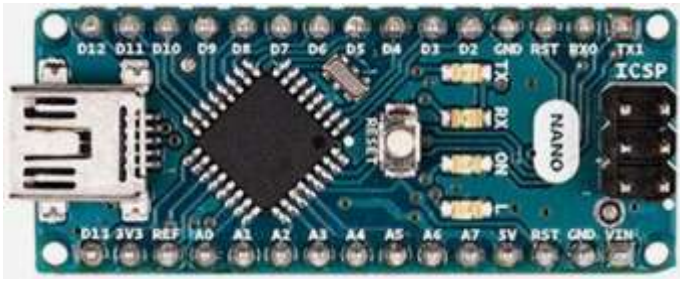
Fonte: https://www.embarcados.com.br/*Arduino*-yun/

O *Arduino* Yún é composto pelo chip *ATmega32u4* que faz o controle das entradas digitais e analógicas e também pelo chip *Athernos* AR 9331 que faz a interface *Wi-Fi*, Ethernet, *MicroSD* e *USB Host*.

2.4.5 *ARDUINO NANO*

Monteiro Neto (2017), destaca que o *Arduino Nano* é um pequeno e completo dispositivo, baseado no micro controlador ATmega328. Ele possui uma entrada de alimentação mini-B USB para conectar ao computador, 14 portas digitais de entrada/saída, 6 portas PWM e 8 portas analógicas. Possui memória de 32 KB, das quais 2 KB é usada pelo *bootloader*. Opera com uma tensão de 5V e pode receber até 40 mA. (Figura 12).

Figura 12 - *Arduino Nano*



Fonte: https://store.*Arduino*.cc/usa/*Arduino*-nano

2.5 MÓDULOS

Para expandir as funcionalidades do *Arduino*, existem módulos chamado de Shields, que são placas adicionais para desenvolvimento que expandem a capacidade da placa *Arduino*, o Shield fica encaixado através de pinos em cima do *Arduino* sem interferir no funcionamento do mesmo, a seguir veremos alguns exemplos de Shields.

2.5.1 *ETHERNET SHIELD* W5100

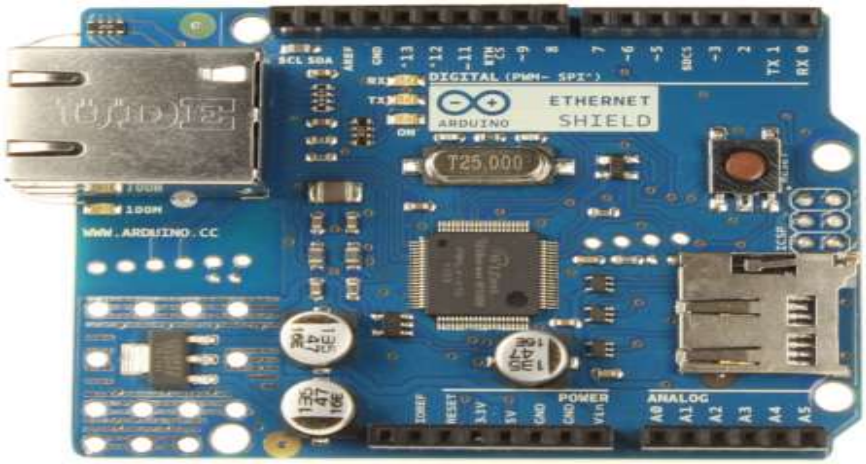
O *Ethernet Shield* (Figura 13), é uma placa de expansão para *Arduino*, ele tem uma entrada para cabo RJ45(cabo de rede), o mesmo que acompanha um roteador de acesso à internet, que possibilita o *Arduino* comunicar em uma rede local ou na internet. Além disso o *Ethernet Shield* possui leitor para cartão SD, para que possa gravar dados em cartão de memória.

É importante observar que ao utilizar uma *Ethernet Shield* no *Arduino*, algumas portas digitais serão reservadas exclusivamente a ele, ou seja algumas portas não poderão mais ser utilizadas para outras tarefas, as portas digitais reservadas são as 4, 10, 11, 12, 13, o restante das portas digitais e analógicas podem ser utilizadas normalmente, sendo assim restam 7 portas digitais: 2, 3, 5, 6, 7, 8, 9 e 6 portas analógicas: A0, A1, A2, A3, A4, A5.

As especificações da *Ethernet Shield* W5100 são:

- Tensão de operação 5V (fornecida pela placa *Arduino*);
- Controlador *Ethernet*: *W5100* com buffer interno 16K;
- Velocidade de conexão: 10/100Mb;
- Conexão com o *Arduino* na porta SPI;
- *Slot* para cartão micro-SD;
- Suporta até 4 conexões simultâneas;
- Compatível com *Arduino* Uno e Mega;

Figura 13: *Ethernet Shield*



Fonte: https://store.*Arduino*.cc/usa/*Arduino*-ethernet-shield-2

2.5.2 SHIELD WIFI ESP8266

O *Shield* ESP8266 (Figura 14), é uma placa de expansão exclusiva para utilização de redes Wireless e tem o funcionamento bem parecido com a *Ethernet Shield*, bastando apenas encaixar o mesmo em cima do *Arduino* UNO. As especificações principais são:

- Regulador de tensão *AMS1117* 3.3V, recebe 5v e converte para 3.3v.
- *WiFi* nativo padrão 802.11b/g/n.
- Compativel com *Arduino* Uno, Mega, Leonardo e Derivados.
- Opera em modo AP, *Station* iou AP + *Station*.

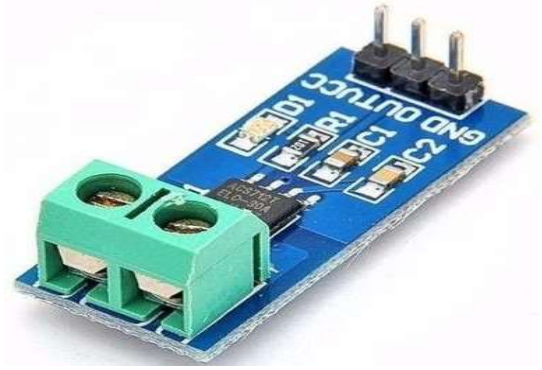
Figura 14 - *Shield WiFi ESP8266*

Fonte: https://i.stack.imgur.com/1Qtcc.jpg

2.5 SENSORES

Sensores são dispositivos eletroeletrônicos que tem a propriedade de transformar em sinal elétrico a transformação de uma grandeza física que está relacionada a uma ou mais propriedades do material de que é feito o sensor" (STEFFENS, 2016). Existem diversos sensores para ser utilizado na plataforma *Arduino* por exemplo: Sensor de Corrente (Figura 15), Sensor de Presença (Figura 16), de Temperatura e Umidade e Sensor Gás e Fumaça. Faz parte deste projeto os sensores de temperatura e umidade AM 3202 *DHT22* e o sensor de fumaça *MQ-2*.

Figura 15 - Sensor de Corrente ACS712 -30A a +30A



Fonte: https://www.vidadesilicio.com.br/sensor-de-corrente-acs712-30a

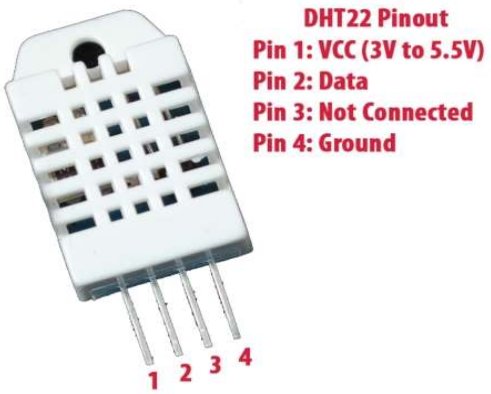
Figura 16 - Sensor de Presença

Fonte: http://mekhos.com.br/mekhos/produto/sensor-de-presenca-pir/

2.6.1 SENSOR *DHT 22*

O sensor *DHT 22*, figura 17, é um pequeno dispositivo digital que funciona integrando a tecnologia de dois sensores: temperatura e umidade, retornando valores com a range de detecção de umidade de 0% a 100%, e de -40 a 80º Celsius para temperatura, sua acurácia possui uma margem de erro de apenas 2% para medições de umidade e 0.5 ºC, para temperatura. O *DHT22* possui quatro pinos de conexão: um para receber energia (VCC), outro para transmitir as informações de medições (Data), um nulo (Não utilizado) e outro para saída (GND).

Figura 17 - Sensor *AM2302 DHT22*



Fonte: https://hobbytronics.com.pk/product/*DHT22*-digital-temperature-and-humidity-sensor/

2.6.2 SENSOR *MQ-2*

O Sensor *MQ-2* é um dispositivo da série MQ capaz de identificar concentrações de gás e fumaça no ar, é muito utilizado em projetos de *Arduino*, pois é um módulo simples e confiável. O sensor de gás *MQ-2* utiliza um pequeno aquecedor no interior com um sensor eletroquímico. Eles são sensíveis para uma

gama de gases e são utilizados em ambientes fechados à temperatura ambiente. (*ARDUINO*, 2005).

O sensor *MQ-2* é capaz de identificar outras substâncias como GLP (gás liquefeito de petróleo), butano e metano, existem diversos modelos de sensores na série MQ como pode ser vista na tabela 03.

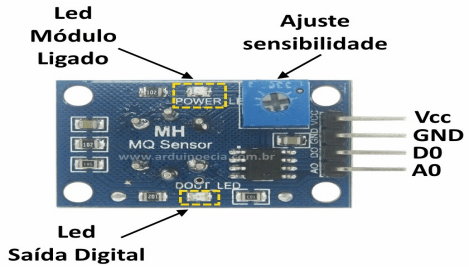
Tabela 03 – Características dos sensores da série MQ

Sensor	Sensibilidade	Tensão aquecimento
MQ-2	Metano, butano, gás GLP, CO, fumo.	5v
MQ-3	Álcool, etanol, fumo.	5v
MQ-4	Metano, gás CNC.	5v
MQ-5	Gás GLP.	5v
MQ-6	Gás GLP, butano.	5v
MQ-7	Monóxido de carbono.	Tensão alternada de 5 e 1.4v
MQ-8	Gás hidrogênio.	5v
MQ-9	Monóxido de carbono, gases inflamáveis.	Tensão alternada de 5 e 1.5v
MQ-131	Ozônio.	6v
MQ-135	Benzeno, álcool, fumo.	5v
MQ-136	Sulfeto de hidrogênio.	5v
MQ-137	Amoníaco	5v
MQ-138	Benzeno, tolueno, álcool, acetona, propano, gás formaldeído, gás hidrogênio.	5v
MQ-214	Metano, gás natural.	6v
MQ-216	Gás natural, gás de carvão.	6v
MQ-303A	Álcool, etanol e fumo.	0.9v
MQ-306A	GLP, gás butano.	0.9v
MQ-307A	Monóxido de carbono.	Tensão alternada de 0.2 e 0.9v
MQ-309A	Monóxido de carbono, gases inflamáveis.	Tensão alternada de 0.2 e 0.9v

Fonte: https://jualabs.wordpress.com/2016/11/30/sensoriamento-de-gases-em-tempo-real-atraves-de-sensores-mq-0-9/

Neste trabalho foi escolhido o sensor *MQ-2*, pois o mesmo além de identificar a presença de fumaça, que é o foco desse trabalho, também consegue identificar outros tipos de gases como Metano, butano, GLP (Gás Liquefeito de Petróleo) e CO. Na figura 18 mostram o sensor e suas características:

Figura 18 - Sensor *MQ-2*



Fonte: https://www.*Arduino*ecia.com.br/2015/01/alarme-sensor-de-gas-modulo-*MQ-2*.html

2.7 DISPLAY LCD 20 X 4 C/ I2C

O Display LCD 20x4 é muito utilizado em projetos com *Arduino* e em equipamentos industriais, ele pode ser operado tanto em 4 como em 8-bits. A vantagem desse display é o tamanho que é maior, facilitando a visualização e por conter mais linhas de textos. A sua ligação com o *Arduino* é simples, pois este modelo vem com um módulo chamado I2C que facilita a conexão, com apenas 4 jumpers é possível realizar a comunicação com *Arduino*, ao invés de 12 jumpers sem o I2C.

Esse display conecta-se ao *Arduino* através dos pinos VCC, GND que devem ser ligados em 5v e terra, e os pinos SDA e SCL que devem ser conectados a porta analógica A4 e A5 respectivamente. Além disso, existe um potenciômetro em que se pode ajustar o brilho do display, bastando apenas girar o mesmo com utilização de uma chave de venda. Também é possível desligar totalmente a luz do display com a retirada de um jumper que fica localizada na parte traseira do sensor, neste display o jumper normalmente vem fechado por padrão. (Figura 19).

Figura 19 - Display LCD 20X4 c/ I2C

Fonte: https://www.tdegypt.com/product/20x4-2004a-lcd-display-with-i2c/

3 AVALIAÇÃO DO *DATA CENTER*

Neste capítulo foi analisada a atual situação da infraestrutura física, da refrigeração e da segurança dos equipamentos que ficam localizados no *Data center*, antigo CPD (centro de processamento de dados) da Justiça Federal, na cidade de Picos/Piauí e constatou-se que o *Data center* atual deixa a desejar nos seguintes aspectos:

3.1 SEGURANÇA FÍSICA

Não há política de acesso às dependências do data center. Para se chegar à sala dos servidores, passa-se por sala de trabalho da Secretaria, o que não impede que qualquer agente administrativo adentre ao local, já que não há autenticação por biometria ou cartões, apenas uma porta comum, conforme a figura 20:

Figura 20: porta de acesso ao data center

Fonte: autor, 2018

3.2 SISTEMA DE CLIMATIZAÇÃO

Foram analisadas as dependências do *Data center* e constatou-se que ainda se utilizam ares-condicionados comuns, e não os de precisão, como os recomendados pelas normas internacionais ANSI/TIA-942. Foi observado também que não havia sistema de monitoramento de temperatura e umidade presente. O sistema de

climatização é realizado por 2 ares-condicionados do tipo “janela” de 12 mil BTUs, conforme a figura 21:

Figura 21: ar-condicionado tipo “janela” de 12 mil BTUs

Fonte: autor, 2018

3.3 SISTEMA DE DETECÇÃO DE FUMAÇA E COMBATE A INCÊNDIO

No *Data center* avaliado, há uma unidade de extintor de incêndio, porém não havia sistema de detecção de fumaça, o que pode dificultar a identificação de um princípio de incêndio e perda de tempo nas ações de contenção do fogo para a preservação da vida, dos equipamentos e das informações contidas no *Data center*. (Figura 22).

Figura 22: sala de servidores e o extintor de incêndio

Fonte: autor, 2018

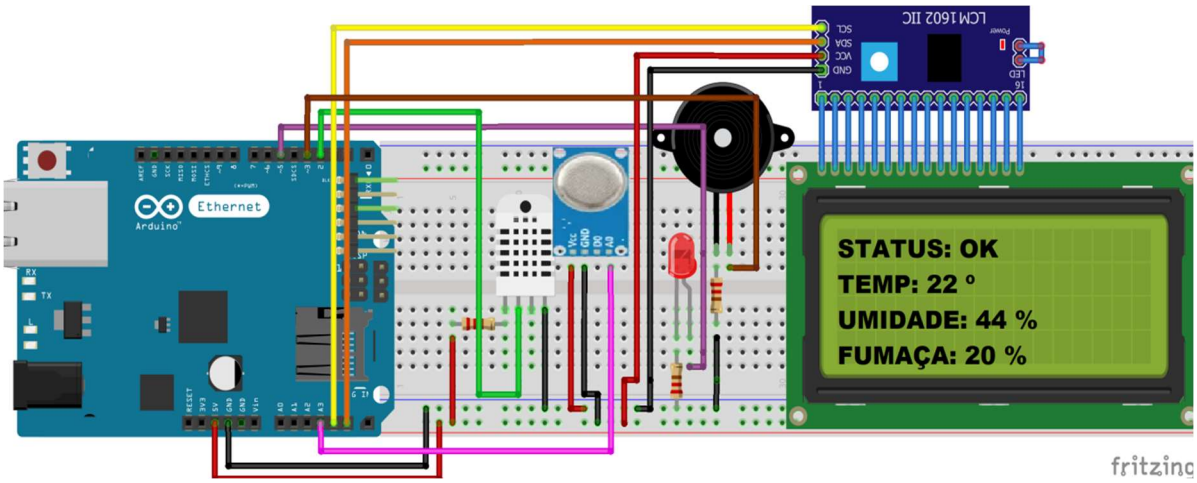
Assim ficou evidente a necessidade de um sistema de monitoramento de temperatura, umidade e de fumaça para alertar possíveis problemas ocasionados por falta de infraestrutura necessária no *Data center* ou defeitos ocasionais que levem a perigos iminentes.

4 DISCUSSÕES E RESULTADOS

Este capitulo destina-se ao projeto de análise e desenvolvimento do sistema proposto, que será realizado em duas etapas, sendo a primeira com montagem e desenvolvimento do protótipo *Arduino* para monitoramento de temperatura, umidade e fumaça bem como a sua codificação. A segunda etapa será realizada a instalação e configuração dos softwares utilizados para obter os dados dos sensores.

Na primeira etapa foi utilizado os softwares: *Arduino* IDE e o Fritzing e os componentes eletrônicos: *Arduino* UNO R3, *Ethernet Shield*, Sensor *DHT22*, Sensor *MQ-2*, *Display* LCD 20x4, uma *protoboard*, *jumpers*, um *buzzer* e um *led*. Primeiramente foi realizado o desenho do projeto utilizando o software *Fritzing*, a sua utilização foi bastante simples, basta arrastar os componentes necessários e fazer a sua ligação, na figura 23 mostra o esquema proposto. É importante observar que a *Ethernet Shield* fica acoplado exatamente em cima do *Arduino* UNO, por este motivo não é possível visualizar o *Arduino* UNO pelo desenho realizado no Fritzing.

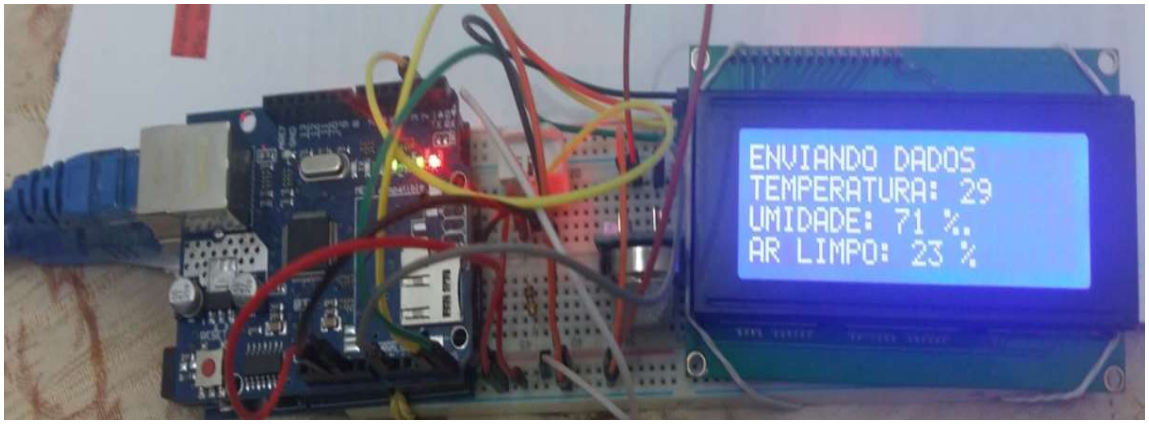
Figura 23. Esquema de ligação do *Arduino* e seus componentes.



Fonte: autor, 2018.

Após o desenho foi realizada a montagem do protótipo, conforme a figura 24, em seguida, o protótipo foi montado numa *case* de plástico para melhorar a estética e dar mais firmeza. (Figura 25).

Figura 24. Protótipo do *Arduino* montado na protoboard.



Fonte: autor, 2018.

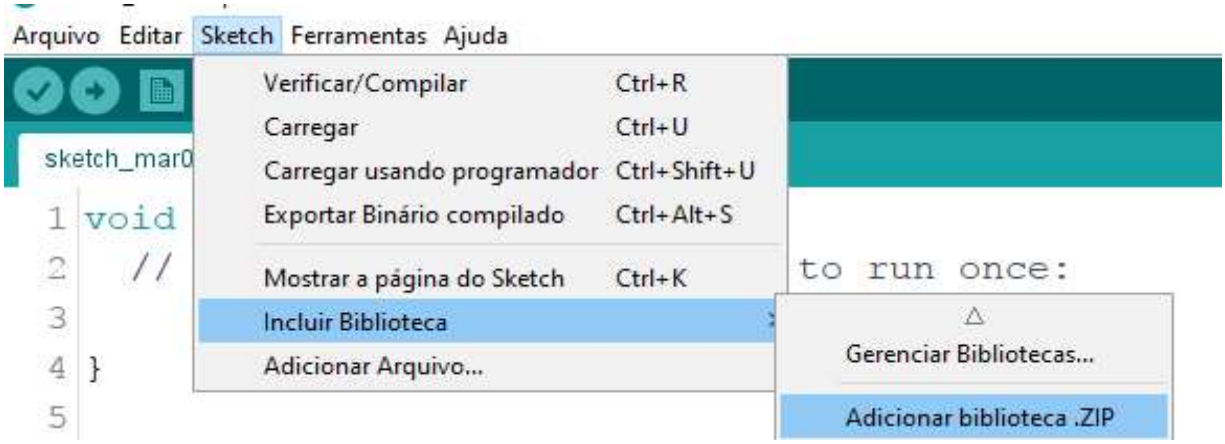
Figura 25. Protótipo do *Arduino* montado numa *case*.

Fonte: autor, 2018.

4.1 DESENVOLVIMENTO DO SOFTWARE

Para desenvolvimento do software, foi utilizado o *Arduino* IDE 1.8.5, um ambiente de desenvolvimento específico para projetos de *Arduino*, antes de realizar a codificação foi feito o *download* das bibliotecas do sensor DHT disponível no repositório *GitHub*: github.com/markruys/*Arduino*-DHT/ e também a biblioteca do *Display* LCD disponível em bitbucket.org/fmalpartida/new-liquidcrystal/. No menu *Sketch* da IDE, foi selecionado a opção incluir biblioteca e em seguida foi localizada e adicionada as bibliotecas necessárias. (Figura 26).

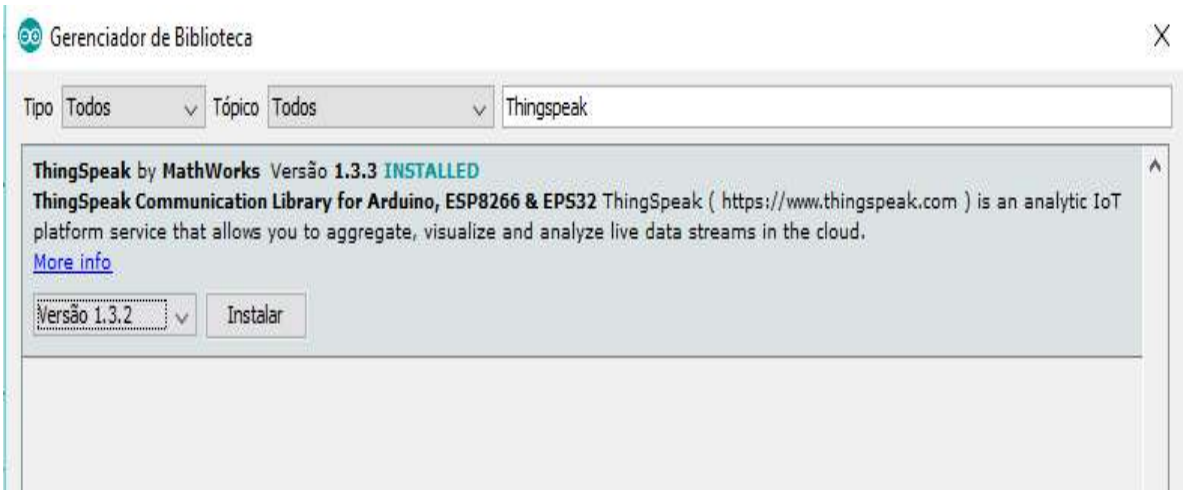
Figura 26. Incluindo bibliotecas necessárias no *Arduino* IDE



Fonte: autor, 2018.

Para o protótipo comunicar com o *Thingspeak,* foi necessário baixar e instalar a biblioteca gratuita do *Thingspeak,* disponível no próprio repositório do Arduino IDE, essa biblioteca é necessária para o funcionamento correto no envio dos dados, para instalar basta ir no menu *Sketch* e depois em gerenciar bibliotecas, em seguida basta localizar pelo nome e clicar em instalar, nesse projeto foi utilizado a versão 1.3.2 conforme a figura 27.

Figura 27. Incluindo bibliotecas pelo gerenciador no *Arduino* IDE



Fonte: autor, 2018.

O segundo passo foi incluir as bibliotecas no código, primeiramente incluímos a biblioteca do sensor de temperatura e umidade *DHT22*, depois definimos o tipo de DHT e o pino em qual ele está conectado. Em seguida incluiu-se as bibliotecas da *Ethernet Shield* para realizar comunicação com a rede, depois foi inserida a biblioteca do *Thingspeak* para conseguirmos enviar os dados para a *Web*, por último foi incluída

as bibliotecas do *Display* LCD. Na figura 28 mostra o código com as bibliotecas inseridas.

Figura 28. Incluindo bibliotecas no código

Fonte: autor, 2018.

No próximo passo foi feito os ajustes dos sensores, display, *Ethernet Shield*, do *Thingspeak* e criação de variáveis necessárias para o funcionamento do sistema conforme a figura 29:

Figura 29. Criação de variáveis necessárias

```
16 //Definindo endereço de MAC
17 byte W5100_MacAddress[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED};
18 //Ajustes no LCD
19 LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);
20 //Iniciando o DHT...
21 DHT dht(DHTPIN, DHTTYPE);
22 //Informações da Conta do Thingspeak
23 unsigned long Channel_Number = 368133; // Canal de acesso
24 const char * Write_API_Key = "5ZPUOILAONDBEI7F"; // Chave de acesso
25 //Campos criados no Thingspeak para receber os dados
26 int Field_Temperatura = 1;
27 int Field_Umidade = 2;
28 int Field_Fumaca = 3;
29 //Inicia o objeto
30 EthernetClient client;
```

Fonte: autor, 2018.

Em seguida foi declarado algumas variáveis necessárias para o funcionamento do sensor de fumaça, do *led* e do *buzzer*. (Figura 30).

Figura 30. Incluindo variáveis necessárias

```
32 //sensor de fumaça MQ-2 conectado na porta analógica A0, não precisa de biblioteca
33 int sensorMq2 = A0;
34 //variavel utilizada para receber os dados do sensor MQ-2 após conversão
35 int valor = 0;
36 //variavel utilizada para controlar led
37 int pin_led = 5;
38 //variavel utilizada para controlar buzzer
39 int pin_buzzer = 6;
```

Fonte: autor, 2018.

Após a criação das variáveis, foi criada a função responsável por iniciar a conexão de rede, ou seja para obter um IP da rede, essa função irá mostrar no *display* o endereço de IP, Máscara de Sub-rede e o *Gateway* (figura 31):

Figura 31. Incluindo funções necessárias

```
41 void conectarRede(){
42   // Conectar a rede
43   Ethernet.begin(W5100_MacAddress);
44   lcd.setCursor(2, 0);
45   lcd.print("REDE DETECTADA!");
46   lcd.setCursor(2, 1);
47   lcd.print(Ethernet.localIP());
48   lcd.setCursor(2, 2);
49   lcd.print(Ethernet.subnetMask());
50   lcd.setCursor(2, 3);
51   lcd.print(Ethernet.gatewayIP());
52   delay(2000);
53 }
```

Fonte: autor, 2018.

Na Função *void setup(),* foi configurado os itens necessários para serem carregados toda vez que o *Arduino* for iniciado, foi definido o modo de operação do sensor, do *led* e do *buzzer* e em seguida chama a função para conectar à rede. (Figura 32).

Figura 32. Incluindo configurações e funções no void setup().

```
57  //inicia LCD de 20 colunas e 4 linhas
58  lcd.begin (20, 4);
59  //Definindo pino A0 como entrada
60  pinMode(sensorMq2, INPUT);
61
62  //definindo led e buzzers como saida
63  pinMode(pin_led, OUTPUT);
64  pinMode(pin_buzzer, OUTPUT);
65  conectarRede();
```

Fonte: autor, 2018.

Na *função void loop()*, foi configurado os itens que irão se repetir até o *Arduino* ser reiniciado. Primeiramente foi feita a leitura dos sensores e um teste para identificar falha nos sensores (Figura 33), caso os mesmos estejam funcionando corretamente será feito um teste de condições para verificar se os dados estão fora da faixa, se caso estiver, será emitido um som através do *buzzer* e o *led* irá piscar para alertar ao usuário. Figura 34:

Figura 33. Incluindo configurações e funções no *void loop().*

```
69 void loop() {
70   float h = dht.readHumidity();//isso vem da biblioteca do Thingspeak
71   float t = dht.readTemperature();
72   valor = analogRead(sensorMq2); //Faz a leitura da entrada do sensor
73   valor = map(valor, 0, 1023, 0, 100); //Faz a conversão da variável para porcentagem
74   // testa se retorno é valido, caso contrário algo está errado.
75   if (isnan(t) || isnan(h) || isnan(valor))
76   {
77     lcd.setCursor(0, 0);
78     lcd.print("FALHA NO DHT22!");
79   }
80   else
```

Fonte: autor, 2018.

Figura 34. Incluindo configurações no *void loop()*

```
80   else
81   {
82     lcd.setCursor(0, 0);
83     lcd.print("MONITORAMENTO CPD");
84     if(t > 32.5){
85     lcd.setCursor(0, 1);
86     lcd.print("TEMP ALTA: " + String(t));
87     digitalWrite(pin_led, HIGH);
88     digitalWrite(pin_buzzer, HIGH);
89     delay(1000);
90     }else{
91     lcd.setCursor(0, 1);
92     lcd.print("TEMP NORMAL: " + String(t));
93     digitalWrite(pin_led, LOW);
94     digitalWrite(pin_buzzer, LOW);
95     }
```

Fonte: autor, 2018.

No fim do loop é feita uma conexão com o servidor *Thingspeak* e em seguida será enviado os dados de temperatura, umidade e fumaça, um por vez, a cada 15 segundos. (Figura 35).

Figura 35. Conectando ao *thingspeak* e enviando dados

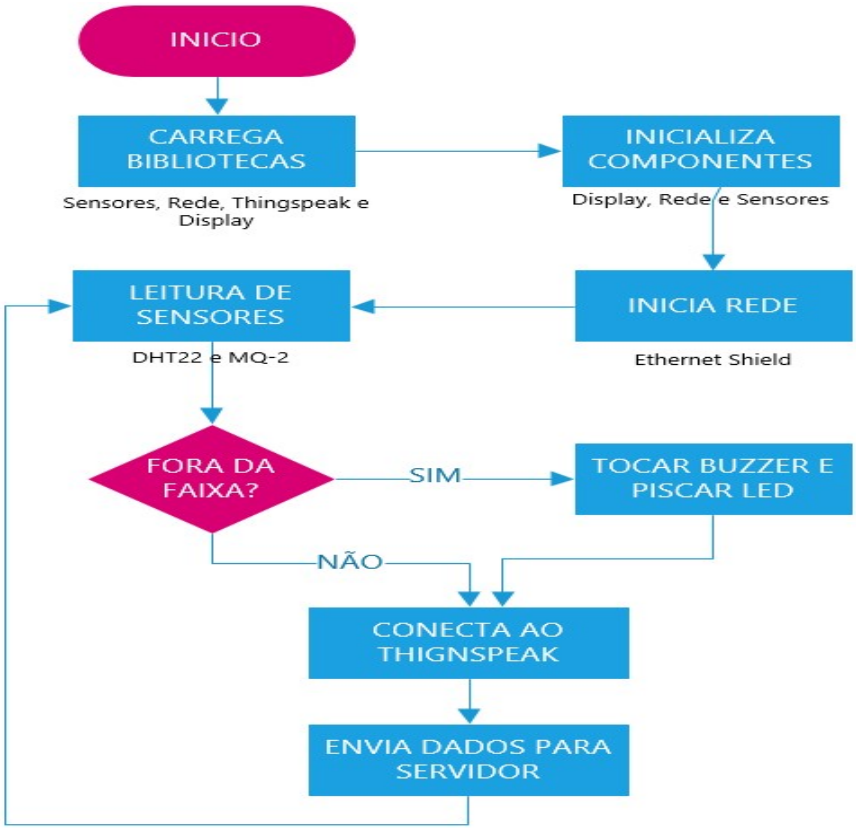
```
124  if (ThingSpeak.begin(client))
125  {
126    lcd.setCursor(0,1);
127    lcd.print("CONECTANDO...");
128  }else{
129    lcd.setCursor(2, 1);
130    lcd.print("FALHA AO CONECTAR");
131  }
132 // enviando dados para um thingspeak, um por um, com um delay de 15 segundos.
133 if (ThingSpeak.writeField(Channel_Number, Field_Temperatura, t, Write_API_Key))
134  {
135    lcd.setCursor(0, 0);
136    lcd.print("TEMP ENVIADO...");
137  }else{
138    lcd.setCursor(0, 0);
139    lcd.print("ERRO ENVIAR DADOS");
140  }
```

Fonte: autor, 2018.

Após a codificação, basta compilar o código e aguardar até a IDE concluir o envio das instruções, após a codificação e a compilação o *Arduino* já estará em funcionamento, coletando e enviando os dados para a plataforma *Web*, faltando

apenas criar a conta na plataforma *Web* para visualizar os dados. Para entender melhor o funcionamento do sistema de proposto foi feito um fluxograma (Figura 36).

Figura 36. Fluxograma do sistema de monitoramento.



Fonte: autor, 2018.

4.2 CRIAÇÃO DA CONTA E CANAIS NO *THINGSPEAK*

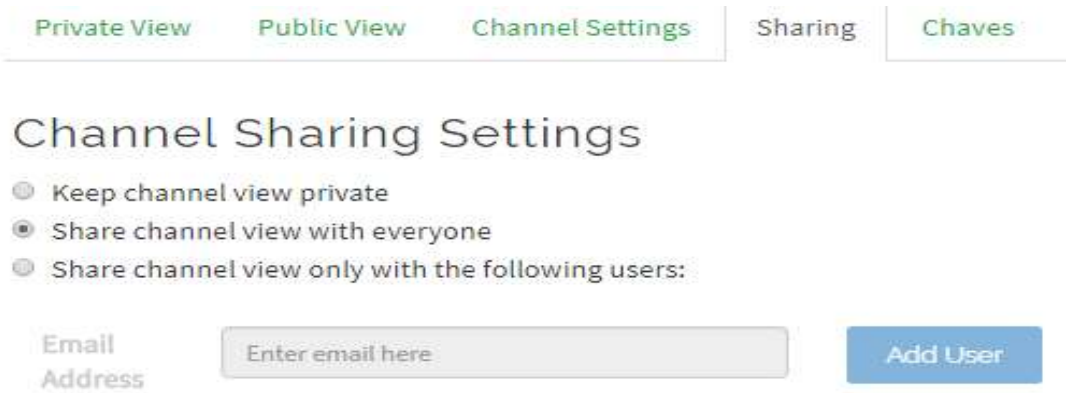
Inicialmente o usuário deve criar uma conta para acesso ao *Thingspeak*, para isso é necessário acessar o site https://thingspeak.com e realizar o cadastro utilizando um e-mail, feito isso, na tela seguinte já temos a opção de criar um canal no repositório, basta clicar em "*New Channel*", preencher os dados de nome, descrição e marcar os campos 1, 2 e 3. Esses campos devem ser preenchidos respectivamente pela pelo nome das variáveis que receberão as informações dos sensores, que no neste caso é temperatura, umidade e fumaça. Após esses ajustes o *Thingspeak* está pronto para receber os dados. Na figura 37 mostra um exemplo demonstrando como criar e configurar o canal.

Figura 37. Criando conta e configurando canal no *Thingspeak*

Fonte: autor, 2018.

Para permitir que qualquer usuário possa acessar o canal, ou seja, para torna-lo público, é necessário configurar o compartilhamento, para isto deve-se ir na opção compartilhando (*Sharing*), e em seguida marcar a opção “Compartilhe a visão do canal com todos” (*Share channel view with everyone*), conforme a figura 38. Após essa configuração, basta copiar a URL com número do canal e compartilhar com público, por exemplo: www.thingspeak.com/channels/368133.

Figura 38. Configurando compartilhamento de canal no *Thingspeak*.



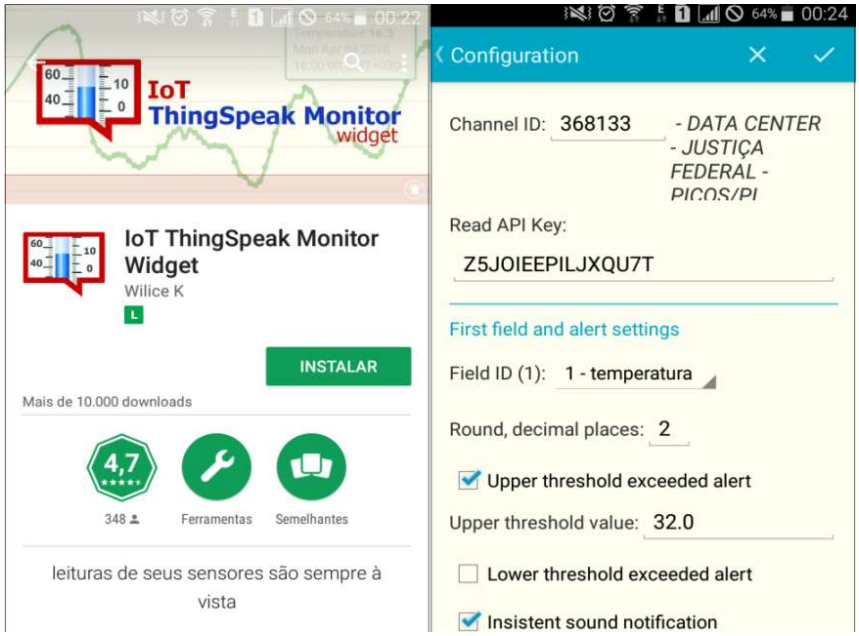
Fonte: autor, 2018.

4.3 INSTALAÇÃO E CONFIGURAÇÃO DA APLICAÇÃO MOBILE

Após criar a conta e configurar o *Thingspeak* deve se instalar o aplicativo para melhorar a visualização dos dados em Smartphones, para isso utilizou-se o aplicativo mobile gratuito *IoT Thingspeak Monitor Widget*, o mesmo encontra-se disponível para *Android*, neste aplicativo é possível visualizar os dados em tempo real através de gráficos e receber alertas por vibração e alarme sonoro, caso os dados estejam em níveis críticos. Este aplicativo fica fixado no papel de parede do *Android*, dessa forma fica mais fácil de visualizar os dados, sem a necessidade de ficar abrindo o aplicativo toda vez.

Para instalar e configurar o aplicativo, o usuário deve primeiro fazer o download na Play Store e, após a instalação, devem-se fazer os ajustes nas configurações, em *Channel ID*, deve-se digitar o número do canal e em *Read API Key* informar a chave do *Thingspeak* (no Menu chaves) e depois escolher o campo (*Field*) que deseja visualizar. Em seguida, deve-se definir um valor limite para que o aplicativo alerte por meio de vibração e alarme sonoro caso o valor identificado seja maior do que o máximo permitido, a fim de avisar o usuário para tomar providencias antes que a situação se agrave. Na figura 39, explica-se como configurar essa funcionalidade.

Figura 39 - Configurando app mobile *IoT Thingspeak Monitor Widget*



Fonte: autor, 2018.

Após a instalação e os ajustes, o aplicativo estará pronto, mostrando os valores de temperatura, umidade e fumaça, (Figura 40). É importante observar que neste aplicativo, o qual atualiza as informações a cada 1 minuto em média para receber os dados, há uma pequena limitação de tempo para receber os dados. Caso o valor definido esteja em níveis críticos, ele irá alertar emitindo som e vibração.

Figura 40 – App *Thingspeak Monitor* em funcionamento

Fonte**:** autor, 2018.

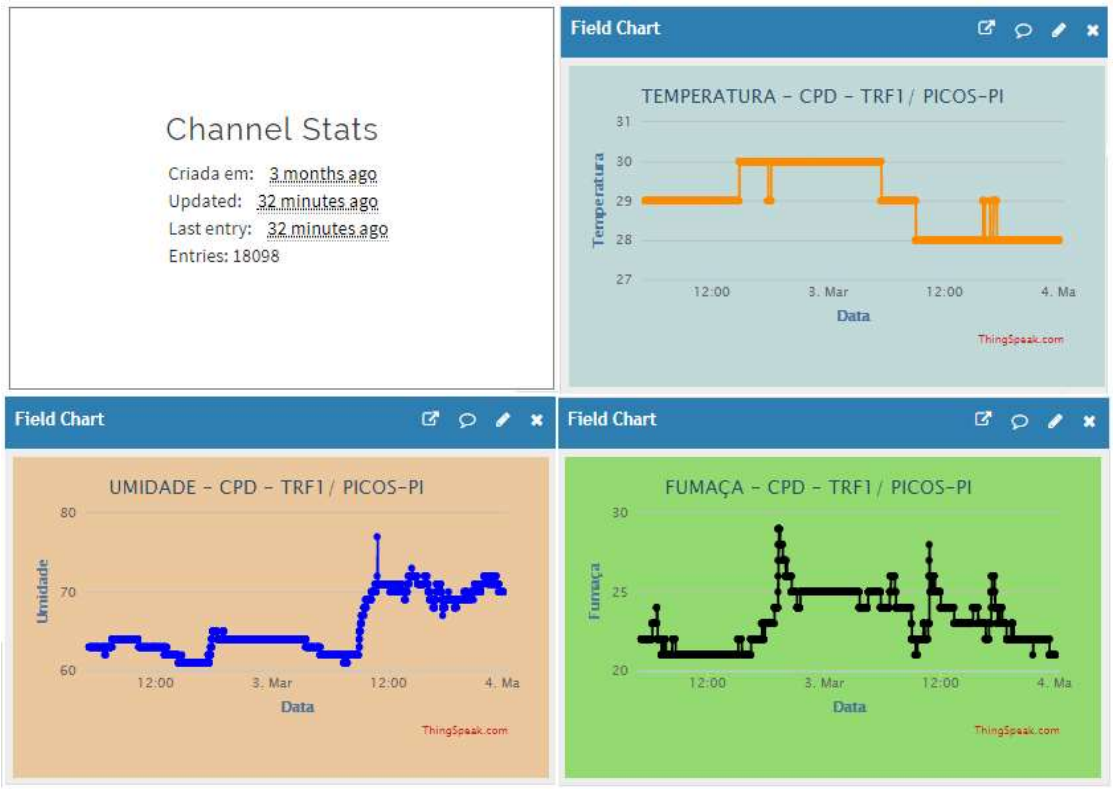
Existem outras opções disponível para recebimento de dados e de alertas, uma delas é utilizando uma API nativa do *Thingspeak*, chamada *ThingTwitter*, que envia alertas através rede social *Twitter* para o *App mobile*, porem o gestor responsável pelo *Data center* considerou essa opção inviável, pois via regra é proibido o uso de redes sociais na instituição pesquisada.

4.4 RESULTADOS

A análise do sistema desenvolvido foi realizada primeiramente em um ambiente simulado nos dias 02 e 03 de março de 2018. Durante a análise o *Thingspeak* recebeu 18.098 dados dos sensores DHT e MQ-2. Não foram observadas falhas em nenhum

dos dois sensores durante o período em que foi monitorado, na Figura 41 mostra os gráficos com a média de temperatura que é medida por graus Celsius e a umidade e fumaça que são medidas por porcentagem.

Figura 41 – Resultados do monitoramento



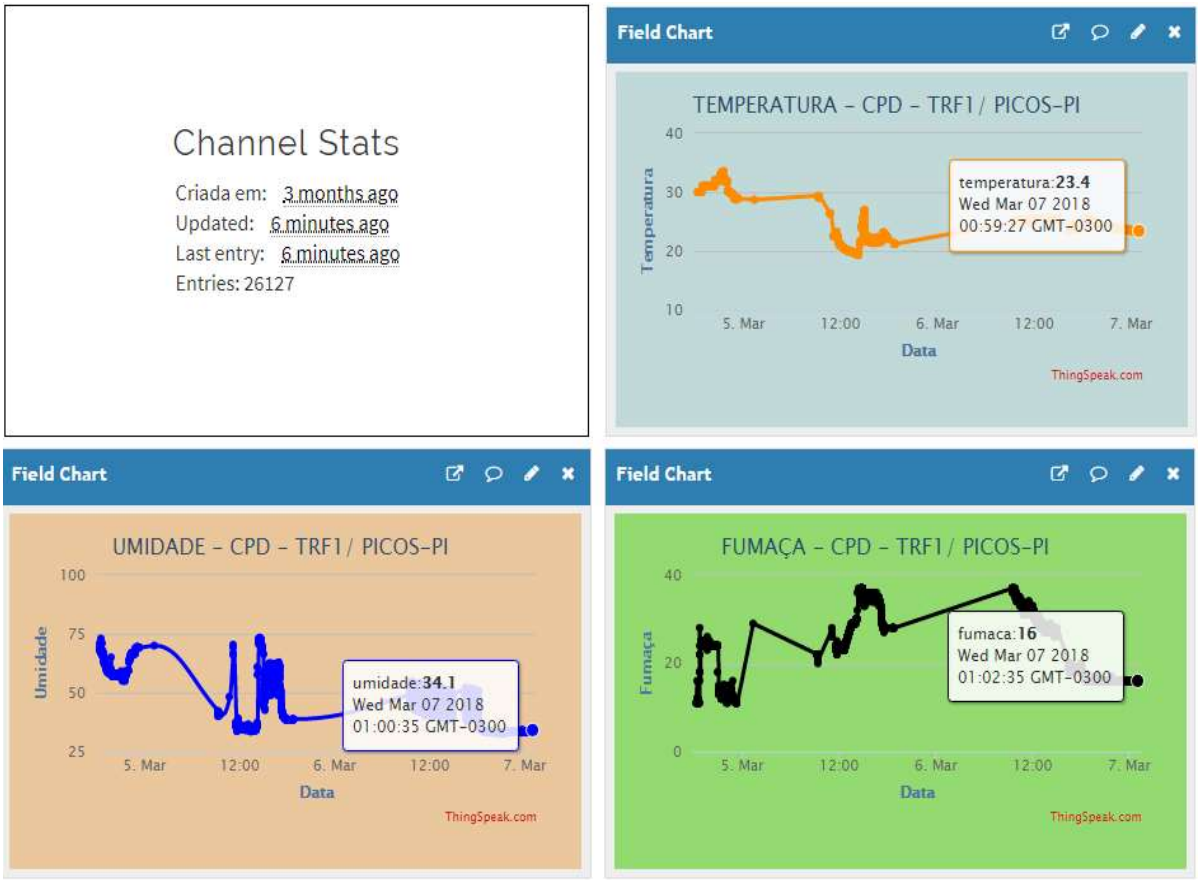
Fonte: autor, 2018.

A segunda análise foi realizada no próprio *data center* da Justiça Federal de Picos. O local escolhido para fixação do protótipo foi no rack central do data center, Figura 43, pois é onde se localizam os servidores de grande poder de processamento de dados e de tráfego de rede, esses equipamentos são considerados críticos e aquecem muito, por isso necessita de prioridade.

Após a instalação, foram registrados 26.127 dados dos sensores nos dias 05 e 06 de março de 2018, e também não foram observadas nenhuma falha na leitura dos sensores durante o período de testes, demonstrando uma alta confiabilidade dos sensores *DHT22* e *MQ-2,* conforme a figura 42, o protótipo foi programado para alertar conforme as normas recomendadas, com valor de temperatura máxima de 25 graus Celsius e 55% de umidade e a fumaça foi programada de acordo com o valor médio

identificado no data center que foi de 30%, caso o valor ultrapasse os 35% o protótipo alertará.

Figura 42 – Resultados do monitoramento



Fonte: autor, 2018.

Figura 43 – protótipo instalado no rack

Fonte: autor, 2018.

4.5 DIFICULDADES ENCONTRADAS

Uma das dificuldades encontradas nesse trabalho foi à pesquisa sobre os *Data centers*, pois a maioria dos artigos e documentos pesquisados eram em inglês, sendo necessária a tradução de boa parte dos documentos. Outra dificuldade foi codificar o *Arduino* para enviar os dados para o repositório *Thingspeak*, pois no primeiro momento percebeu-se que os dados às vezes não chegavam no repositório, outras vezes chegavam com um grande atraso.

Depois de algumas pesquisas chegou-se à conclusão que o *Thingspeak* só permite recebimento de dados a cada 15 segundos, sendo necessário acrescentar um "*delay*" de 15 segundos antes de cada linha do código responsável por enviar os dados, dessa forma o problema foi solucionado.

5 CONCLUSÃO

Esta monografia apresentou o que seria necessário para o desenvolvimento de um dispositivo de internet das coisas para monitoramento de temperatura, umidade relativa e fumaça por meio da placa *Arduino* UNO em que seria automatizado o processo de coleta e o envio desses dados para um sistema *web*, facilitando assim a visualização deles a partir de qualquer dispositivo com acesso à internet. Com isso seria possível não apenas realizar o monitoramento constante desses dados para prever e alertar possíveis anormalidades na temperatura e umidade, como também detectar sinais de fumaça.

Devido à necessidade de a Justiça Federal realizar o monitoramento ambiente do *Data center* e aos elevados custos desses equipamentos disponíveis no mercado para atender a essa demanda, o desenvolvimento de uma solução baseada em *hardware* de baixo custo e *open source* é uma ótima alternativa para redução dos custos. Na concepção oficial do gestor responsável pelo *Data center*, este projeto possui um potencial bastante interessante para ser adotado no dia a dia.

O objetivo do trabalho foi atingido no que diz respeito ao desenvolvimento e a sua implantação. Além disso, ao longo do desenvolvimento, foi adicionada uma solução a qual não havia sido inicialmente prevista, que foi o monitoramento de fumaça do ambiente.

Não obstante os objetivos do presente trabalho terem sido satisfatoriamente atendidos, as implementações e melhorias em ambientes de missão crítica, como os *Data centers*, não param de acontecer. Portanto, dada à relevância do tema e complexidade das tecnologias estudadas, faz-se necessário um aprofundamento constante sobre o assunto.

Como possibilidade de trabalhos futuros, sugere-se desenvolver um sistema web exclusivo, com visualização dos dados em gráficos e relatórios para integrar à intranet da Justiça Federal, facilitando ainda mais a visualização dos dados. Outra sugestão interessante seria realizar a integração do sistema desenvolvido com o *Zabbix,* um sistema de monitoramento de rede utilizado pela Justiça Federal que também pode monitorar os dados do *Arduino*. A maior vantagem dessa integração é a centralização do monitoramento de todos os dispositivos de rede da Justiça Federal em um único sistema.

REFERÊNCIAS

ADAMI, Anna. ***DATA CENTER***, [entre 2006 a 2018] <https://www.infoescola.com/internet/data-center/>. Acesso em 21 Mar 2018.

ATZORI, Luigi; IERA, Antonio; MORABITO, Giacomo. **The Internet of Things: A survey,** 2010. Computer Networks 54 (2010), p. 2787–2805. Disponível em: <http://www.elsevier.com/__data/assets/pdf_file/0010/187831/The-Internet-ofThings.pdf>. Acesso em: 19 de Mar. 2018.

ANSI/TIA-942. **Telecomunicantios infrastructure standard for *DataCenters***. Arlington, USA: TIA, 2005.

ARDUINO. **MQ Gas sensors**. 2005. Disponível em: <http://playground.*Arduino*.cc/Main/MQGasSensors>. Acesso em: 07/03/2018.

ARDUINO. ***Arduino Ethernet Shield***. Disponível em: <https://www.*Arduino*.cc/en/Main/*Arduino*EthernetShield.> Acesso em: 08 jan. 2018.

CAMHI, Jonathan, **BI Intelligence projects 34 billion devices will be connected by 2020.** <http://www.businessinsider.com/bi-intelligence-34-billion-connected-devices-2020-2015-11>. Acesso em 08 de mar 2018.

CARRAHER, D. W. **Senso crítico: do dia-a-dia às ciências humanas**. São Paulo: Cengage Learning, 2008.

DRIEMEYER, R. **Projeto De Melhoria De Data Center Com Ênfase Em Infraestrutura E Eficiência Energética.** 2016. 19. Monografia. UNIVATES. Lajeado, 2016.

FILHO, Mauro Faccioni. **Gestão da Infraestrutura do Datacenter**. Livro Digital, Universidade do Sul de Santa Catarina – Unisul, Palhoça, 2016.

FREITAS DIAS, R. R. **Internet das coisas sem mistérios: uma nova inteligência para os negócios**. São Paulo: Netpress Books, 2016.

FRIZZARIN, F. B. **Arduino Guia para Colocar Suas Ideias em Prática** . São Paulo: Casa do Código, 2016.

GATTI, B. A. **A construção da pesquisa em educação no Brasil**. Brasília: Líber Livro, in Transition Politics. Princeton: Princeton University Press. 2007. p. 32 Metodologia do Trabalho Científico: aspectos introdutórios / Maria Candida Soares Del-Masso – Marília : Oficina Universitária ; São Paulo : Cultura Acadêmica, 2012

HRIBERNIK, Karl A.; GHRAIRI, Zied; HANS, Carl; THOBEN, K. **First experiences in the participatory design of intelligent products with *Arduino*** (ICE 2011).

MARCONI, Marina e LAKATOS, Eva. Fundamentos de metodologia científica. São Paulo, 5 edition, 2010.

MANYIKA. J. et al. **The *Internet Of Things*: mapping the value beyond the hype. Technical report**, Mckinsey Gobal Institute, 2015.

MARKETSANDMARKETS. **Middleware para Internet das Coisas (IoT) será mercado de US$ 11, bi em 2020** (2015), disponível em <http://www.voit.com.br/middleware-para-internet-das-coisas-iot-sera-mercado-de-us-11-bi-em-2020/>. Acesso em 20 de mar. 2018.

MINAYO, Maria Cecília de Souza. **O desafio do conhecimento**. São Paulo: Hucitec, 1993

MARIN, P. S. ***DataCenters*: desvendando cada passo: conceitos, projetos, infraestrutura física e eficiência energética**. 1. Ed. São Paulo: Érica, 2011.

MARIN, Paulo Sérgio. **Data Centers: desvendando cada passo: projeto, infraestrutura física e eficiência energética**. 1 ed. São Paulo: Érica, 2013.

MOREIRA, Bruno. O Setor Elétrico. ***IoT* é a nova revolução tecnológica**, Santa Cecília, São Paulo, v. 123, n. 11, p. 58-61, Acesso em: jan de 2018

MUKHOPADHYAY, S. C. ***Internet Of Things*, Smart Sensors, Measurement and Instrumentation 9**. Springer Internetional Publishing Switzerland, 2014.

MCROBERTS, Michael. ***Arduino* Básico**. São Paulo, 1 edition, 2011.

OLIVEIRA, N.B, R.Z, T,A. **Aplicações De Automação Em Iot – Internet Of Things.** 2016. 19.

RAMPAZZO, L. **Metodologia cientifica**, 3. Ed. São Paulo: Editora Loyola, 2005.

RIBEIRO, Flavio. **PRTG – monitorando totalmente sua rede e infraestrutura**. (2014) Disponível em <http://www.fxreview.com.br/2012/11/prtg-monitorando-totalmente-sua-rede-e.html>. Acesso em: 20 dez. 2017.

RUYS, Mark. **An *Arduino* library for reading the DHT family of temperature and humidity sensors**. URL <https://github.com/markruys/*Arduino*-DHT>. Acesso em: 09 fev 2018.

SOBRAL, M. M. **Sistema De Monitoramento Remoto De Consumo De Energia E Temperatura Em Racks De Datacenter.** 2015. 73. Monografia. UniCEUB. Brasilia, 2015.

STEFFENS, César A. **Sensores**. Disponível em <http://www.if.ufrgs.br/mpef/mef004/20061/Cesar/SENSORES-Definicao.html>. Acesso em: 07 fev. 2018.

VIEIRA Emily C. **Jaqueta, lâmpada e fechadura: veja exemplos do potencial da Internet das Coisas**. (2013) Disponível em <http://tecnologia.ig.com.br >. Acesso em: 05 fev. 2018.

ZASLAVSKY, A.; PERERA, C. **Context Aware Computing for The Internet of Things: A Survey**. Disponível em: < http://arxiv.org/pdf/1305.0982.pdf>. Acessado em 18 de Mar. 2018.

APÊNDICES

Código 01 – Código Fonte do Sistema de Monitoramento de temperatura, umidade e fumaça (*Arduino*).

```
//Biblioteca do Sensor de Temperatura e Umidade DHT
#include "DHT.h"
//Definir o tipo de DHT, neste caso é o DHT22
#define DHTTYPE DHT22
//Definir a porta digital onde o DHT está conectado
#define DHTPIN 2
//Incluir Bibliotecas da Ethernet Shield
#include <SPI.h>
#include <Ethernet.h>
//Incluir Biblioteca da plataforma web para IoT ThingSpeak
#include "ThingSpeak.h"
//Incluir Bibliotecas do LCD 20x4
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

//Definir endereço de MAC
byte W5100_MacAddress[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED};
//Ajustar LCD
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);
//Iniciar o DHT...
DHT dht(DHTPIN, DHTTYPE);

//Inserir Informações da Conta do Thingspeak
unsigned long Channel_Number = 368133; // Canal de acesso
const char * Write_API_Key = "5ZPUOILAONDBEI7F"; // Chave de acesso
//Campos criados no Thingspeak para receber os dados
int Field_Temperatura = 1;
int Field_Umidade = 2;
int Field_Fumaca = 3;

//Iniciar o objeto
EthernetClient client;

//sensor de fumaça MQ-2 conectado na porta analógica A0, não precisa de
biblioteca
int sensorMq2 = A0;
//variavel utilizada para receber os dados do sensor MQ-2 após conversão
int valor = 0;
//variavel utilizada para controlar led
int pin_led = 5;
//variavel utilizada para controlar buzzer
int pin_buzzer = 6;

void conectarRede(){
  // Conectar a rede
  Ethernet.begin(W5100_MacAddress);
  lcd.setCursor(2, 0);
```

```
  lcd.print("REDE DETECTADA!");
  lcd.setCursor(2, 1);
  lcd.print(Ethernet.localIP());
  lcd.setCursor(2, 2);
  lcd.print(Ethernet.subnetMask());
  lcd.setCursor(2, 3);
  lcd.print(Ethernet.gatewayIP());
  delay(2000);
}

void setup(){

  //inicia LCD de 20 colunas e 4 linhas
  lcd.begin (20, 4);
  //Definindo pino A0 como entrada
  pinMode(sensorMq2, INPUT);

  //definindo led e buzzers como saida
  pinMode(pin_led, OUTPUT);
  pinMode(pin_buzzer, OUTPUT);
  conectarRede();

}

void loop() {
  float h = dht.readHumidity();//isso vem da biblioteca do Thingspeak
  float t = dht.readTemperature();
  valor = analogRead(sensorMq2); //Faz a leitura da entrada do sensor
  valor = map(valor, 0, 1023, 0, 100); //Faz a conversão da variável para
porcentagem
  // testa se retorno é valido, caso contrário algo está errado.
  if (isnan(t) || isnan(h) || isnan(valor))
  {
    lcd.setCursor(0, 0);
    lcd.print("FALHA NO DHT22!");
  }
  else
  {
    lcd.setCursor(0, 0);
    lcd.print("MONITORAMENTO CPD");
    if(t > 32.5){
    lcd.setCursor(0, 1);
    lcd.print("TEMP ALTA: " + String(t));
    digitalWrite(pin_led, HIGH);
    digitalWrite(pin_buzzer, HIGH);
    delay(1000);
    }else{
    lcd.setCursor(0, 1);
    lcd.print("TEMP NORMAL: " + String(t));
    digitalWrite(pin_led, LOW);
```

```
  digitalWrite(pin_buzzer, LOW);
  }
  if(h > 75){
  lcd.setCursor(0, 2);
  lcd.print("UMID ALTA: " + String(h) + " %");
  digitalWrite(pin_led, HIGH);
  digitalWrite(pin_buzzer, HIGH);
  delay(1000);
  }else{
  lcd.setCursor(0, 2);
  lcd.print("UMID NORMAL: " + String(h) + " %");
  digitalWrite(pin_led, LOW);
  digitalWrite(pin_buzzer, LOW);
  }

  if (valor > 30) {
    lcd.setCursor(0, 3);
    lcd.print("ALERTA FUMACA: " + String(valor) + " %");
    digitalWrite(pin_led, HIGH);
    digitalWrite(pin_buzzer, HIGH);
    delay(1000);
  } else {
    lcd.setCursor(0, 3);
    lcd.print("AR LIMPO: " + String(valor) + " %");
    digitalWrite(pin_led, LOW);
    digitalWrite(pin_buzzer, LOW);
  }
}

// Conectando ao servidor thingspeak
 if (ThingSpeak.begin(client))
 {
   lcd.setCursor(0,1);
   lcd.print("CONECTANDO...");
 }else{
   lcd.setCursor(2, 1);
   lcd.print("FALHA AO CONECTAR");
 }

if (ThingSpeak.writeField(Channel_Number, Field_Temperatura, t, Write_API_Key))
 {
   lcd.setCursor(0, 0);
   lcd.print("TEMP ENVIADO...");
 }else{
   lcd.setCursor(0, 0);
   lcd.print("ERRO ENVIAR DADOS");
 }
 delay(15000);

 if (ThingSpeak.writeField(Channel_Number, Field_Umidade, h, Write_API_Key))
```

```
  {
    lcd.setCursor(0, 0);
    lcd.print("UMID ENVIADA...");
  }else{
    lcd.setCursor(0, 0);
    lcd.print("ERRO ENVIAR DADOS");
  }
  delay(15000);

  if (ThingSpeak.writeField(Channel_Number, Field_Fumaca, valor, Write_API_Key))
  {
     lcd.setCursor(0, 0);
     lcd.print("FUMAC ENVIADA...");
  }else{
    lcd.setCursor(0, 0);
    lcd.print("ERRO ENVIAR DADOS");
  }
  delay(15000);
}
```