



**Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"**

GUSTAVO DOS SANTOS DINIZ

**ESTAÇÃO METEOROLÓGICA PORTÁTIL UTILIZANDO OS DISPOSITIVOS
ARDUINO E PYTHON COM A PLATAFORMA MQTT**

**Assis/SP
2018**



**Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"**

GUSTAVO DOS SANTOS DINIZ

**ESTAÇÃO METEOROLÓGICA PORTÁTIL UTILIZANDO OS DISPOSITIVOS
ARDUINO E PYTHON COM A PLATAFORMA MQTT**

Projeto de pesquisa apresentado ao curso de Ciências da Computação do Instituto Municipal de Ensino Superior de Assis – IMESA e a Fundação Educacional do Município de Assis – FEMA, como requisito parcial à obtenção do Certificado de Conclusão.

**Orientando(a): Gustavo dos Santos Diniz
Orientador(a): Prof. Douglas Sanches da Cunha**

**Assis/SP
2018**

FICHA CATALOGRÁFICA

DINIZ, Gustavo dos Santos.

**ESTAÇÃO METEOROLÓGICA PORTÁTIL UTILIZANDO OS
DISPOSITIVOS ARDUINO E PYTHON COM A PLATAFORMA**

MQTT / Gustavo dos Santos Diniz. Fundação Educacional do
Município de Assis

–FEMA – Assis, 2018.35.

1. Internet das coisas. 2. Acesso remoto.3 Sensores.4 Automação

CDD:005.304
Biblioteca da FEMA

ESTAÇÃO METEOROLÓGICA PORTÁTIL UTILIZANDO OS DISPOSITIVOS ARDUINO E PYTHON COM A PLATAFORMA MQTT

GUSTAVO DOS SANTOS DINIZ

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, avaliado pela seguinte comissão examinadora:

Orientador:

Prof. Douglas Sanches da Cunha

Examinador:

Prof. Fernando Cesar de Lima

**Assis/SP
2018**

AGRADECIMENTOS

Queria agradecer primeiramente a Deus, que me deu a oportunidade de viver, e fez com que eu não pensasse em desistir em nenhum momento, que fez possível a realização de um sonho que é a graduação. Aos meus pais Eva lima dos Santos Diniz e Juliano Cícero Diniz, além da Legião Mirim e do Rotary Club que me deram uma bolsa durante todo o curso. Sou extremamente grato por meu professor e orientador Douglas Sanches da Cunha por todo o suporte e apoio sobre a monografia. A todos os educadores que passaram na minha vida durante esses anos, por me passarem todo o conhecimento que conquistei até hoje e a meus amigos e familiares que nunca me deixaram desistir dos meus sonhos.

DEDICATÓRIA

Dedico este trabalho a Deus, a meus pais a meus amigos, professores, Rotary Club, Legião Mirim, ao meu orientador que fez ser menos cabeça dura, ao Luciano Cristiano de Oliveira Braun, minha filha por me motivar e minha faculdade como um todo.

“Deixem que o futuro diga a verdade e avalie cada um de acordo com o seu trabalho e realizações. O presente pertence a eles, mas o futuro pelo qual eu sempre trabalhei pertence a mim” .

Nikola Tesla

RESUMO

Resumo: Este artigo tem como objetivo apresentar uma maneira de captar dados de uma estação meteorológica a distância fazendo com que tal pessoa comum possa ter o controle de tudo de sua própria casa ou local de trabalho podendo assim ter seu próprio banco de dados e fazer seus próprios testes e cálculos locais.

Utilizando os conceitos de Internet das coisas em conjunto com outras diversas plataformas, se iniciará uma busca de obter uma forma de captar as informações do tempo utilizando objetos baratos como Sensores.

A automação é um fator crucial para que o projeto tenha seu devido sucesso e claro visando sempre o Acesso Remoto para que o projeto chegue a sua conclusão de acordo com a expectativa inicial.

Palavras-chave: Internet das coisas, Sensores, Automação, Acesso Remoto.

ABSTRACT

Summary: This article aims to present a way of capturing data from a weather station in the distance doing with such a common person can be in control of everything from your own home or workplace and can therefore have your own database and do their own tests and calculations locations.

Using the concepts of Internet of things in conjunction with several other platforms, will begin a search for a way to capture weather information using objects, cheap Sensors.

Automation is a crucial factor for the project to have its due success and of course always aiming at Remote Access so that the project reaches its conclusion according to the initial expectation.

Keywords: Internet of Things, Sensors, Automation, Remote Access.

Lista de Ilustrações

Figura 1. Arduino Uno R3. (ARDUINO, 2018).....	16
Figura 2. Sensor de Água (Core Technologies);.....	18
Figura 3. Sensor de Luz (ldr) (Core Technologies).....	19
Figura 4. Sensor de Temperatura/Umidade (DHT11) (Core Technologies).....	19
Figura 5. PROTOBOARD (ROBOCORE).....	20
Figura 6. Logo Eclipse. (Eclipse.org, 2018).....	21
Figura 7. Broucker Cute Cat. (CloudMQTT, 2018).....	23
Figura 8. Arquitetura do Projeto, Do Autor.....	25
Figura 9. Painel de Controle do CloudMQTT.....	26
Figura 10. Configurações do Broucker.....	26
Figura 11. Configuração da conexão do Arduino com o programa Python.....	27
Figura 12. Configuração da conexão da aplicação com o MQTT.....	28
Figura 13. Interface Desktop.....	29
Figura 14. Protótipo Aberto.....	30
Figura 15. Protótipo Pronto.....	30

LISTA ABREVIATURAS E SIGLAS

IOT	<i>Internet of Things</i> (Internet das coisas)
RAM	<i>Random access memory</i> (Memória de acesso aleatório)
SD	<i>Secure digital</i>
USB	<i>Universal serial bus</i> (Barramento serial universal)
GPIO	<i>General purpose input/output</i> (Entradas/saídas de propósitos gerais)
FULL HD	<i>Full high definition</i> (Alta definição total)
IDE	<i>Integrated Development Environment</i>
MQTT	<i>Message Queue Telemetry Transport</i>

Sumário

LISTA ABREVIATURAS E SIGLAS.....	11
1. INTRODUÇÃO.....	13
2. ARDUINO.....	15
2.1 SENSORES.....	17
2.2 SENSOR DE ÁGUA.....	17
2.3 SENSOR DE LUZ (LDR).....	18
2.4 SENSOR DE TEMPERATURA/UMIDADE DHT11.....	19
2.5 PROTOBOARD.....	20
3. ECLIPSE.....	21
6. DESENVOLVIMENTO DO TRABALHO.....	24
CONCLUSÃO.....	32
REFERÊNCIAS.....	33

1. INTRODUÇÃO

Meteorologia é uma ciência que estuda a atmosfera, que foca nos processos atmosféricos e nas interações dentre seus estados dinâmicos, físico e químico, com a superfície terrestre implícita. Essa ciência é uma das mais importantes existentes no planeta terra pois ela trata de tentar prever o que vai acontecer futuramente não só para as pessoas no seu dia mas também para os agricultores que necessitam muito saber sobre o tempo.

Se soubermos quando vai chover, saberemos quando utilizar o guarda-chuva, quando devemos partir em viagem, quando devemos plantar, construir muito mais.

É importante prever a falta de chuva, a sobra de chuva, períodos de geadas, entre outros diversos fatos que levaram a escolha desse tema.

Atividades econômicas em geral dependem disso afinal o tempo é quem comanda tudo e se conseguíssemos prever com exatidão com certeza o mundo seria outro.

No contexto geral objetivo principal é estudar e desenvolver um protótipo que consiga fazer o controle, coleta, armazenamento e tratamento de dados atmosféricos a fim de conseguir prever o tempo diretamente do conforto de onde estiver.

Pretende-se, com o presente trabalho, implementar um dispositivo que faça a previsão do tempo utilizando o Arduino e fazendo com que ele junto aos sensores se conecte com uma interface gráfica envie as informações coletadas e as trate diretamente de um smartphone com o fim de:

- Compreender as características do cenário da Meteorologia
- Pesquisar e explorar o protocolo MQTT
- Pesquisar a IDE ECLIPSE em conjunto com a linguagem de programação Python
- Modelar e implementar um protótipo de IoT utilizando o protocolo MQTT
- Utilizar diversos sensores em conjunto com o Arduino

No momento não existem muitos artifícios desenvolvidos nessa área de coleta de dados e tratamento dos dados coletados de uma maneira barata simples e pessoal. Buscar coletar esses dados de uma maneira fácil, simples e rápida é o principal objetivo do protótipo.

Sabendo que no mercado existem diversos sensores entre outras diversas coisas conectadas a internet, é justo e aproveitável que tudo seja utilizado para que a

tecnologia avance e ajude os humanos em todos os sentidos não só no meio agrícola mas em todos os possíveis.

Como primeiro passo para futuros trabalhos na área esse dispositivo ficará responsável por dar o ponta pé inicial nessa área fazendo com que vários outros venham a ser criados.

Por meio de estudos e pesquisa na área da meteorologia não se encontra aplicações ou programas acessíveis e baratos utilizando Arduino a fim de automatizar a previsão do tempo, e visar a acessibilidade para qualquer pessoa .Contribuir para área meteorologia vem ser uma das prioridades dessa pesquisa podendo unir assim outras áreas em um só projeto com outras tecnologias integradas onde novas pesquisas futuramente sejam geradas para resolver ou criar soluções para problemas relacionados ou não ao assunto. Baratear custo e criar algo que possa vir a ajudar pessoas é muito importante para que essa pesquisa e seu desenvolvimento sejam de grande e real destaque.

A tecnologia que temos hoje em dia nos ajuda com tudo no cotidiano facilitando a vida humana em tudo, utilizar essas tecnologias simples, mas eficientes na área meteorológica vai contribuir para que outras pessoas também tomem a iniciativa de criar coisas parecidas ou até mesmo aprimorar projetos existentes para que a humanidade chegue ao ponto em que todos os conhecimentos sejam bastante e acessível para que todos os problemas climáticos sejam tratados de formas eficazes para todos.

Utilizar das tecnologias como a do Arduino em conjunto com o IOT abrirá portas para que mais estudos sejam feitos. São tecnologias que unidas são capazes de criar aplicativos, projetos e dispositivos uteis a sociedade e por ser um desafio grande juntá-los faz com que mesmo que inicialmente seja complicado juntar os mesmos faz com que essa experiência seja empolgante pelo fato de querer muito chegar em um resultado final eficiente.

O projeto busca pesquisar sobre o tempo, usar a placa do Arduino com componentes e sensores para que o projeto realmente se concretize.

Esse universo é pouco explorado, portanto estudar em livros, artigos e vídeos foram atos extremamente importantes para conclusão da pesquisa e da finalização do projeto.

A linguagem que utilizada para interagir com o Arduino foi o PYTHON.

2. ARDUINO

Arduino é uma placa eletrônica flexível, simples de usar, econômica e open source, destinada a profissionais e curiosos em geral.

É projetado com um microcontrolador Atmel AVR , sua linguagem de programação tem origem em Wiring e é essencialmente C/C++.

O Arduino é uma plataforma eletrônica de código aberto baseada em hardware e software fáceis de usar. Placas Arduino são capazes de ler entradas de luz em um sensor, um dedo em um botão ou uma mensagem no Twitter e transformá-lo em uma saída ativando um motor, ligando um LED, publicando algo online.

O Arduino também simplifica o processo de trabalho com microcontroladores, mas oferece algumas vantagens para professores, alunos e amadores interessados em outros sistemas: Barato as placas Arduino são relativamente baratas em comparação com outras plataformas de microcontroladores. A versão mais barata dos módulos do Arduino pré-montados custam menos de US \$ 50.

Plataforma cruzada o software Arduino (IDE) é executado nos sistemas operacionais Windows, Macintosh OSX e Linux. A maioria dos sistemas de microcontroladores é limitada ao Windows.

Ambiente de programação simples e claro o Arduino Software (IDE) é fácil de usar para iniciantes, mas flexível o suficiente para os usuários avançados aproveitarem também. Para os professores, ele é convenientemente baseado no ambiente de programação Processing, para que os alunos que aprendem a programar nesse ambiente estejam familiarizados com o funcionamento do Arduino IDE.

Software open source e extensível o software Arduino é publicado como ferramentas open source, disponível para extensão por programadores experientes. A linguagem pode ser expandida através de bibliotecas C ++, e as pessoas que querem entender os detalhes técnicos podem dar o salto do Arduino para a linguagem de programação AVR C na qual ele se baseia. Da mesma forma, você pode adicionar o código do AVR-C diretamente em seus programas do Arduino, se quiser.

Hardware open source e extensível, os planos das placas Arduino são publicados sob uma licença Creative Commons, então designers de circuitos experientes podem fazer sua própria versão do módulo, estendendo-o e aprimorando-o. Até mesmo usuários relativamente inexperientes podem construir a versão de breadboard do módulo para entender como funciona e economizar dinheiro(Arduino,2018).

Em resumo ele é uma pequena placa de processamento de circuitos elétricos que não tem um foco específico em questão de software ou hardware. Realiza seus processos elétricos com dispositivos

físicos (protoboard, resistores, capacitores, jumpers, relés, Leds, displays, botões, etc.

Seguindo esse caminho é possível realizar diversas aplicações com o mesmo independente do que se quer construir existem várias ferramentas que ajudam inclusive em milhares de projetos relacionados a IOT, podendo acoplar diversas (shields) de uso específico para que siga a necessidade de qualquer projeto, dando funcionalidades extras a base inicial de um arduino comum(ARDUINO,2018).

No projeto em questão o Arduino Uno foi o escolhido para o projeto, existem milhares de outros modelos de Arduino, inclusive por ser um hardware livre qualquer um pode inventar sua própria placa.

Na imagem abaixo podemos ver uma placa Arduino com componentes e entradas e saídas a amostra.



Figura 1. Arduino Uno R3. (ARDUINO, 2018)

2.1 SENSORES

Sensores são dispositivos que respondem a estímulos específicos, podendo assim ser utilizados em monitoramento, e medição.

Existem milhares de tipos de sensores mas nesse protótipo foram utilizados três sensores diferentes, sensor de água, luz e temperatura/umidade.

Os sensores estão ao nosso redor. O mundo está cheio deles: de sensores infravermelhos em detectores de movimento a detectores de monóxido de carbono nas residências, até mesmo em, minúsculos acelerômetros, módulos GPS e câmeras dentro de seu smartphone. Graças à proliferação dos dispositivos móveis, os sensores atualmente estão incrivelmente acessíveis quanto ao preço, o que significa que você pode combinar sensores de baixo custo com placas baseadas em microcontroladores para criar os seus próprios dispositivos (Kimmo Karvinen e Tero Karvinen, 2014).

2.2 SENSOR DE ÁGUA

Esse sensor será o responsável por medir a quantidade de chuva que vai cair no medidor e será enviada para a IDE ECLIPSE analisada pela programação que foi em Python, mostrada na tela ao usuário no desktop e enviada para o celular do mesmo em tempo real. Todas as informações são coletadas pela IDE do Arduino Uno e enviadas para o programa pela porta COM correspondente que no caso é a 9.

Ele mostrara no desktop um valor que irá de 0 a 1024 onde 0 significa seca/falta de chuva e 1024 chuva extrema (Adilson Thonsen, 2013).

Na figura abaixo podemos visualizar o sensor de Água utilizado no projeto:



**Figura 2. Sensor de Água
(Core Technologies);**

2.3 SENSOR DE LUZ (LDR)

Esse sensor será responsável por medir a quantidade de luz do local, ou seja ele mostrará a quantidade de luz solar ou da lua e servirá para mostrar se estará de dia ou de noite ou se estará nublado ou limpo o tempo.

Irá mostrar as informações de 0 a 1024 onde o 0 significará que esta completamente noite e 1024 ensolarado.

As informações dele foram implementadas na IDE do Arduino e serão recebidas pela porta COM 9 pela aplicação Python e será mostrada no Desktop do usuário além de ter suas informações enviadas para o celular do mesmo (Adilson Thonsen,2013).

Na imagem abaixo podemos ver o sensor de luz(ldr) utilizado no projeto:



**Figura 3. Sensor de Luz (ldr)(
Core Technologies).**

2.4 SENSOR DE TEMPERATURA/UMIDADE DHT11

O DHT11 é o sensor escolhido para medir a temperatura e a umidade do local, onde ele vai receber essas informações que ele coleta e vai enviar para a aplicação Python que vai mostrar as informações na tela desktop do usuário e enviar para o celular do mesmo. As informações serão dadas em graus para o usuário e todas essas informações serão enviadas pela porta COM9 (Adilson Thonsen,2013).

Na imagem abaixo podemos visualizar o sensor de umidade/temperatura DHT11:

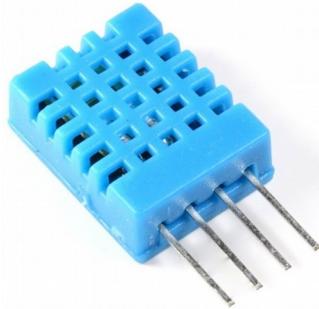


Figura 4. Sensor de Temperatura/Umidade (DHT11) (Core Technologies).

2.5 PROTOBOARD

Ainda se tratando de componentes extras a placa como os sensores descritos acima foi necessária a utilização em conjunto e não acoplada à placa de uma protoboard (ou placa de ensaio). É um componente que pode ser integrado a microcontroladores como o próprio arduino, para que se consiga realizar o desenvolvimento de projetos na eletrônica.

Consiste em uma placa com múltiplos furos que estão ali para servir de barramento de tensão de cobre para os circuitos, podendo ter diversos tipos de funções pra se adequar a diversos tipos de projetos. Para que uma protoboard funcione existem maneiras de ligação como os cabos jumper (condutor elétrico para placas integradas), resistores (limitadores de corrente elétrica) e capacitores de carga (componente armazenador de cargas elétricas de forma a proteger contra sobrecargas sobre um circuito elétrico), variando também o formato da protoboard, quantidade de furos de tensão e variações de voltagem de acordo com o modelo,

fabricante e patente utilizada. Com ela é possível realizar a conexão entre o Arduino para os outros periféricos como os circuitos, leds, displays, sensores, montando assim o esquema físico que será utilizado (WARREN YOUNG.,2011-2016).

Na figura abaixo podemos visualizar uma protoboard:



Figura 5. PROTOBOARD (ROBOCORE).

3. ECLIPSE

A IDE (Intergrated development environment) Eclipse é uma plataforma de desenvolver softwares baseada em Java. Vem com um conjunto padrão de plug-ins para desenvolvimento de softwares, incluindo o JDT que possui ferramentas para desenvolvimento em Java. O Eclipse é escrito em Java, porém não é utilizado apenas para programar na linguagem Java, estão disponíveis para instalação plug-ins para outras linguagens, por exemplo, C/C++ e COBOL. A estrutura da IDE também é usada como base para outros tipos de aplicações não relacionados à criação de software, com sistemas de gerenciamento de conteúdo. Possui um plug-in para trabalhar com Concurrent Versions System (CVS), onde o mesmo se conecta a um servidor CVS e permite que os membros de uma equipe de desenvolvimento trabalhem em um conjunto de arquivos de código de origem sem conflitos com as alterações de outros membros. A IDE possui software livre, ou seja, seu código fonte é aberto. (ANISZCZYK, 2012). O Eclipse é uma IDE que é capaz de compreender várias linguagens através de plug-ins que são instalados e permitem emular o desenvolvimento da plataforma.

Inicialmente foi desenvolvido para o JAVA mas logo ganhou força em dezenas de outras linguagens.

A facilidade de desenvolver nessa IDE a tornou muito popular, ela disponibiliza vários frameworks para desenvolvimento desktop e web, conta com vastas bibliotecas, além de ser simples e intuitiva.

Foi um projeto que se iniciou na IBM, com um gasto inicial de 40 milhões de dólares que de acordo com a própria IBM valeu cada centavo pois hoje em dia é a IDE JAVA mais utilizada em todo o mundo.

Um dos maiores motivos da escolha do mesmo para esse projeto foi por ele ser um software livre, sem custo para o desenvolvedor de pequeno porte (ANISZCZYK, 2012).

Na imagem abaixo podemos ver o ícone do software:



Figura 6. Logo Eclipse. (Eclipse.org, 2018).

4. PYTHON

Python é uma linguagem de alto nível orientada a objeto lançada no ano de 1991 por Guido Van Rossun.

Diversas plataformas estão disponíveis para a linguagem e seu interpretador desde Unix, Linux, FreeBSD, Solaris, MacOS X, Windows. Net, até mesmo alguns celulares, como a série 60, N8xx(pyMaemo) da Nokia.

Para gerar o Python a partir da fonte é necessário que exista um compilador , o código fonte será traduzido pelo interpretador para o formato bytecode, que será multiplataforma onde será executado e distribuído sem fonte original(Python,2018).

Python possui uma implementação muito conhecida chamada CPython , toda em C distribuída com uma biblioteca padrão que mistura C e Python. A implementação é suportada em diversas plataformas, incluindo Unix-Like e Microsoft Windows.

A licença do Python é livre e aprovada pela OSI compatível com a GPL, os binários da linguagem são distribuídos sem a necessidade de fornecer o código fonte junto.

Um programa em Python é feito a partir de blocos de código. O bloco é uma parte do texto programa Python que é executado como uma unidade. Os seguintes são blocos: um corpo de função, um módulo e uma definição de classe. Cada comando digitado interativamente é um bloco. Um arquivo de script (um arquivo fornecido como entrada padrão para o interpretador ou específico como um argumento de linha de comando para o interpretador) é um bloco de código (Python,2018).

5. PROTOCOLO MQTT

O protocolo MQTT foi desenvolvido pela IBM na década de 90,e tinha como objetivo vincular sensores a satélites.

Nas palavras de Michael Yuan (2017) ele expõe o porque de se escolher tal protocolo no meio de tantos.

“O MQTT é um protocolo de rede leve e flexível que oferece equilíbrio ideal para os desenvolvedores de IoT:

O protocolo leve permite a implementação em hardware de dispositivo altamente restringido e em redes de largura da banda limitada e de alta latência.

Sua flexibilidade possibilita o suporte a diversos cenários de aplicativo para dispositivos e serviços de lot.”

De acordo com Gantait (2016), o protocolo MQTT é o mais utilizado para o desenvolvimento de recursos inteligentes em IoT, visto que, é leve e fácil de utilizar com tal protocolo, é possível garantir três tipos de autenticações:

- Autenticação por usuário e senha: Usuário e senha são enviados ao conectar em um *broker* MQTT e ambos não são criptografados pelo protocolo, mas pode ser criptografado no transporte.
- Autenticação com *token* de acesso: O cliente recupera um *token* de acesso e utiliza o mesmo na mensagem de conexão pelo campo de *password*. O tamanho do *token* não pode ultrapassar 65535 bytes, tamanho limite dos campos.
- Autenticação baseada em certificado: O *broker* utiliza de certificados para usar como parte do processo de autenticação mútua.

Esse protocolo teve grande destaque no mundo e foi dado como ideal para esse projeto também por ser gratuito.

Na imagem abaixo temos o *brocker* utilizado no projeto:



Figura 7. Brocker Cute Cat.
(CloudMQTT, 2018).

6. DESENVOLVIMENTO DO TRABALHO

Nesse capítulo vão ser mostradas as experiências e resultados obtidos após o aprofundado estudo sobre Internet das Coisas, Automação e a Linguagem Python durante o trabalho. A arquitetura do projeto apresentada no capítulo anterior foi utilizada como base para a criação da arquitetura final. O protótipo utilizando Arduino está inserido na parte física do projeto, onde é conectado aos sensores de chuva, sensor de temperatura e umidade DHT11 e sensor de luminosidade LDR. A parte sistêmica do projeto é realizada pela IDE ECLIPSE e pela linguagem de programação Python em conjunto com suas bibliotecas e o protocolo MQTT de comunicação.

No projeto prático foi necessário a utilização de duas plataformas para os testes com a plataforma CloudMQTT, e o Python, com o objetivo de assegurar, por meio dos testes, que o *broker* disponibilizado pela empresa seria ideal para a realização do projeto. Com a plataforma Java foi utilizada uma biblioteca chamada Eclipse Paho, que disponibiliza classes e métodos para a conexão com os *brokers*.

Como no capítulo anterior mostrou obtive total sucesso com as plataformas e linguagens escolhidas, se comunicaram perfeitamente e atenderam todas as minhas necessidades de um protótipo inicial. Será apresentado o problema que se pretende resolver, a arquitetura do projeto desenvolvido e quais tecnologias foram adotadas, além de como, juntas, elas se relacionam para resolver o problema proposto.

Utilizando as tecnologias já escritas anteriormente e a proposta a ser realizada é um dispositivo que internet como principal conexão a internet sendo que o Arduino estará conectado aos sensores, tendo seus dados enviados para o celular da pessoa o dia todo e assim facilitando a toda pessoa ter sua própria estação meteorológica.

A definição de um problema se estabeleceu no desenvolvimento de um protótipo inteligente utilizando a IDE ECLIPSE, o protocolo MQTT, Arduino e a linguagem Python. Para o desenvolvimento do trabalho foram utilizados conceitos de Internet das Coisas e do protocolo de comunicação MQTT para a comunicação com o protótipo e os sensores conectados ao Arduino. Os sensores podem ser de diversos estímulos, como luminosidade, temperatura, umidade e chuva, por exemplo. A manipulação e exibição dos dados ficaram sobre responsabilidade da aplicação desktop e por um aplicativo que está na play store, pois possibilita ao programador utilizar de apenas uma linguagem de programação em toda a pilha de ferramentas.

Na Figura abaixo é apresentada a arquitetura do projeto com as tecnologias necessárias para a resolução do problema definido.

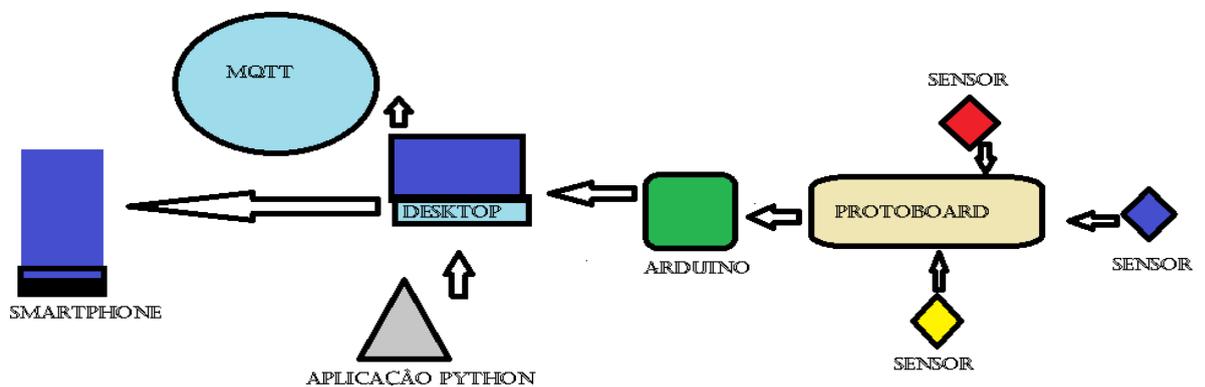


Figura 8. (Arquitetura do Projeto, Do Autor).

Para o desenvolvimento do protótipo é necessária utilização do Arduino com objetivo de comunicar com os sensores e fazer com que ocorra obtenção dos dados por meio do MQTT, um protocolo leve e muito utilizado em projetos de Internet das Coisas. O Arduino foi escolhido principalmente pela facilidade de alterar o seu hardware, pelo preço acessível a grande parte dos iniciantes e fácil integração com diversos sensores. O Python faz a comunicação com o Arduino através de um broker, onde o arduino publica os dados obtidos dos sensores ao broker e a aplicação Python, que está inscrita no broker, recebe esses dados. Para a exibição dos dados, o Python

utiliza sua biblioteca TK e o Canvas, além de um aplicativo próprio MQTT que já se encontra na Play Store. O MQTT será usado na comunicação entre o Arduino conectado aos sensores e a Web API (*Application Programming Interface*). Para que essa comunicação ocorra de forma correta, o protocolo MQTT disponibiliza algo parecido com URLs que são chamados de tópicos (BARROS, 2015). Para que, tanto a API possa receber os dados, quanto o Arduino possa distribuir as informações, deverá haver um tópico em que esse cliente (API) se conecte no caso do protótipo no próprio computador pela porta COM9.

Será utilizado junto a uma solução chamada CloudMQTT, que é um sistema web que disponibiliza um *broker* para aplicações em Internet das Coisas. O Arduino será um publicador (*publisher*), enquanto API será o inscrito (*subscriber*) que receberá os dados publicados pelos *publisher*. O CloudMQTT proporciona maior facilidade na configuração de um *broker* e os tópicos necessários para a comunicação entre publicadores e inscritos.

Primeiramente, para a configuração de um *broker* no CloudMQTT, é necessário criar uma conta na solução, como sugere a documentação (CLOUDMQTT, 2017). Após a criação da conta, é preciso criar uma instância de um *broker* no painel de controle (*Panel Control*), como mostrado na Figura 9.

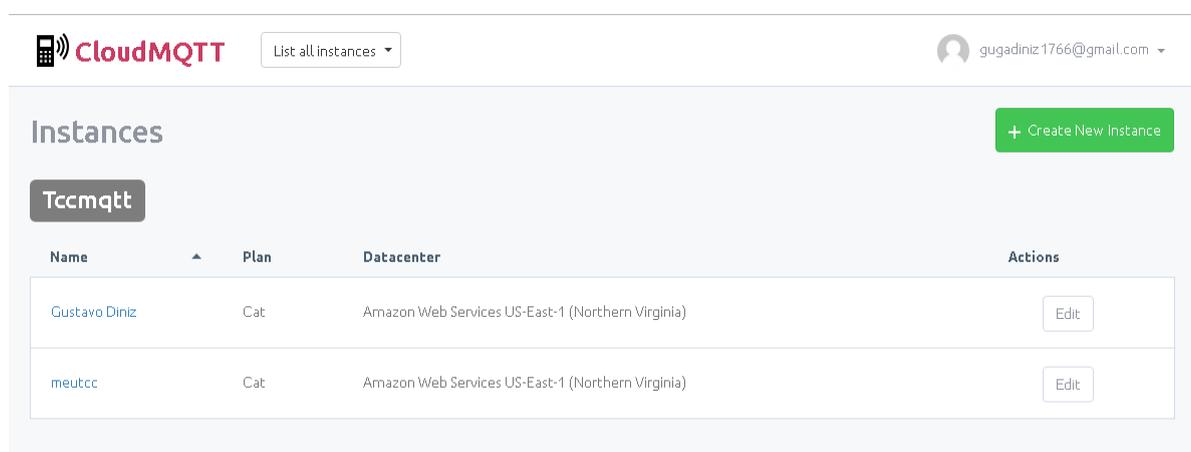


Figura 9.(Painel de Controle do CloudMQTT)

Para acessar as informações como tópicos, usuários e criação de novos tópicos, é necessário ir para os detalhes clicando sobre o nome *meutcc* mostrado na Figura 10.

Na Figura 10 são mostradas configurações como o servidor, usuário, senha, porta e limite de conexões para aquele *broker*.

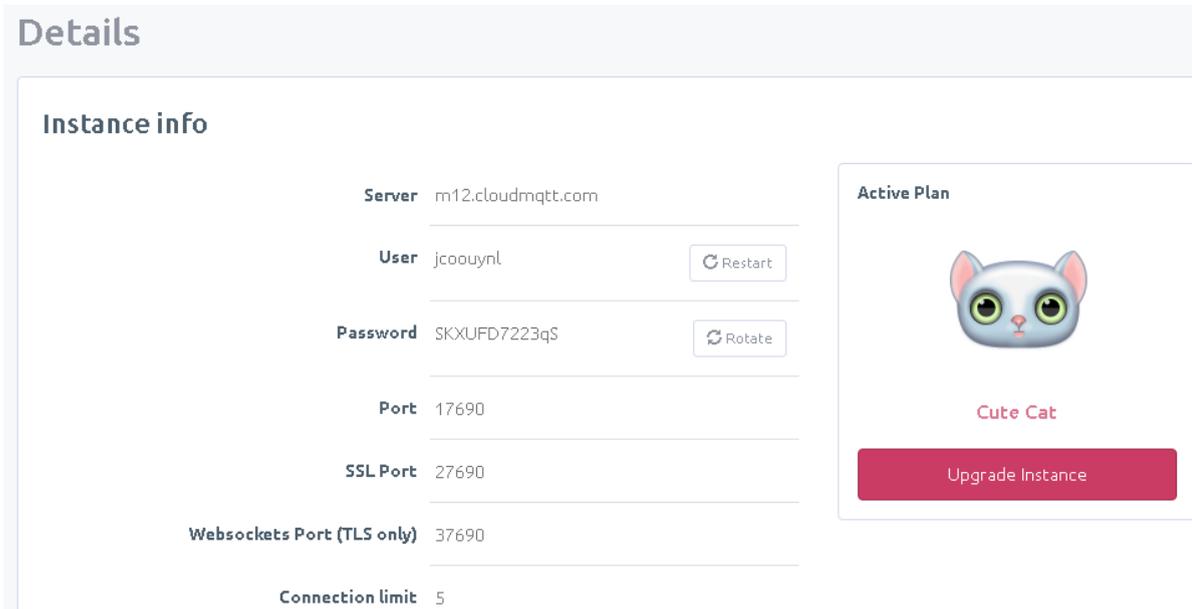


Figura 10.(Configurações do Brocker).

Sempre válido lembrar que nesse broker gratuito não é suportado mais que 5 conexões. Como já dito anteriormente foi usado a linguagem Python para gerar a interface gráfica para o desktop do usuário e para fazer a comunicação com o MQTT. Foi muito utilizada a biblioteca TK para o desenho dos gráficos como o escopo em sim e do Canvas para a formatação das imagens que foram usadas na interface.

Na figura 11 podemos observar como o Arduino se conecta ao código da interface:

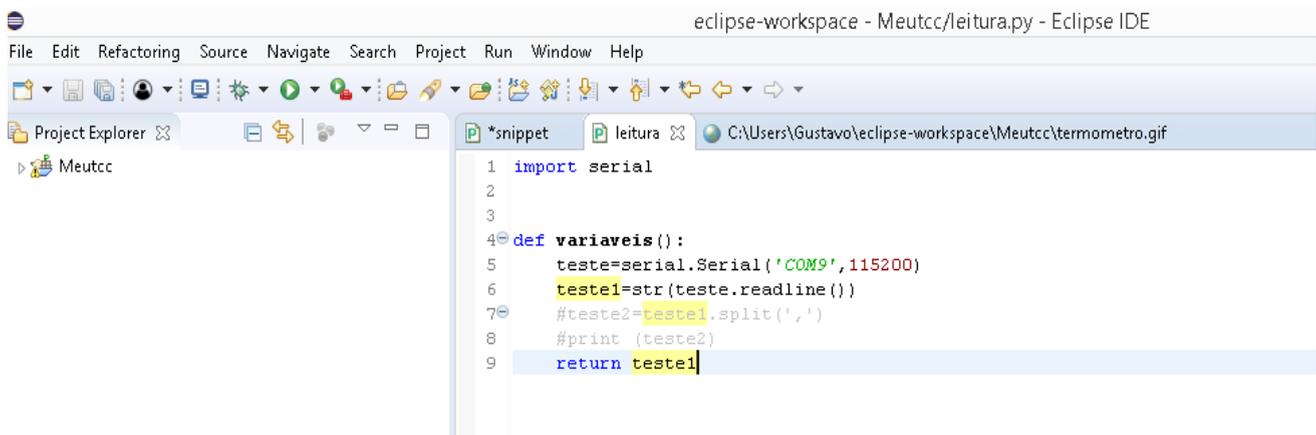


Figura 11. Configuração da conexão do Arduino com o programa Python.

A parte mais importante dessa interface é o código que faz com que ela se conecte ao MQTT e faz com que as informações cheguem para o usuário final. Qualquer erro aqui ocasiona o não funcionamento da aplicação.

Na figura 12 podemos ver esse trecho do código:

```
118 #-----configuração do sqlite-----
119 self.envia="temp:"+str(self.variaveltermometro)+"ar:"+ str(self.variavelhumidade)+"agua:"+ str(self.variavelhumidadeagua)
120 if self.controle==True:
121     arquivo= open('C:\\Users\\Gustavo\\Desktop\\tcc\\parametros.txt', 'r+')
122     self.parametros=arquivo.readline()
123     self.parametros=self.parametros.split(',')
124     publish.single(self.parametros[4],self.envia, hostname=self.parametros[0],port=int(self.parametros[1]),client_id="Diniz",
125                 auth= ('username':self.parametros[2], 'password':self.parametros[3]) )
126 #-----configurando banco de dados sqlite-----
127 self.date=str(datetime.datetime.fromtimestamp(int(time.time())).strftime('%d/%m/%Y %H:%M:%S'))
128 connection=sqlite3.connect('C:\\Users\\Gustavo\\Desktop\\tcc\\mqttcc.db')
129 c=connection.cursor()
130 c.execute('CREATE TABLE IF NOT EXISTS dados(id integer,\\
131 keyword text,datestamp text,temperatura integer, humidadear integer,humidadeagua integer)')
132 keyword= 'estacao meteorologica com mqtt integrado'
133 if self.conta<=100:
134     self.conta+=1
135     c.execute("INSERT INTO dados(id,keyword,datestamp,temperatura,humidadear,humidadeagua) \\
136 VALUES(?,?,?,?,?)", (self.conta,keyword,self.date,self.variaveltermometro,self.variavelhumidade,self.variavelhumidadeagua))
137     connection.commit()
138 else:
139     self.conta2+=1
140     c.execute("""
141 UPDATE dados
142 SET datestamp = ?,temperatura= ?,humidadear = ?,humidadeagua = ?
143 WHERE id = ?
144 """, (self.date, self.variaveltermometro,self.variavelhumidade,self.variavelhumidadeagua, self.conta2))
145     connection.commit()
146
147 if self.conta2>100:
148     self.conta2=0
```

Figura 12. (Configuração da conexão da aplicação com o MQTT).

Com tudo já configurado temos agora a interface pronta para o usuário, e para que ele consiga enviar os dados para o servidor que está no norte da Virgínia ele deve seguir os passos que a imagem 13 exibirá:



Figura 13. (Interface Desktop).

Como a imagem acima mostra a ordem tem que ser exatamente essa, host, porta, nome, senha e tópico tudo separado por vírgulas.

Na figura 14 podemos ver o protótipo montado e encaminhado pro seu término:

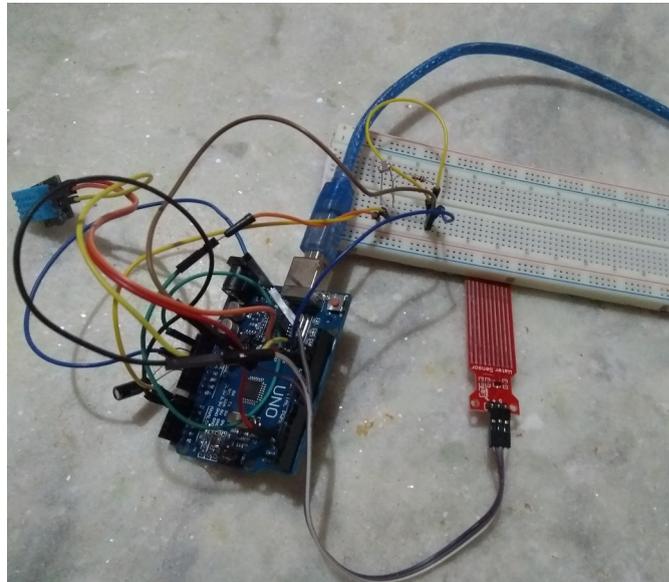


Figura 14. (Protótipo Aberto).

Na figura 15 temos o protótipo já em estado de pronto:

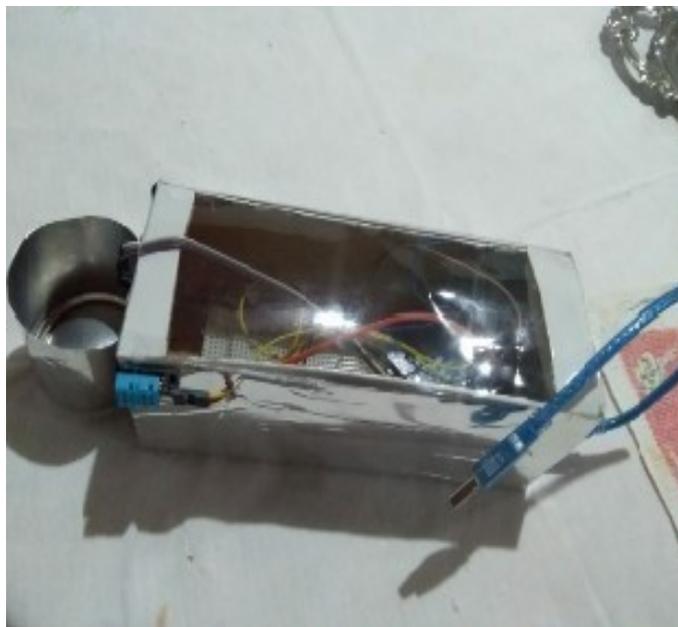


Figura 15: Protótipo Pronto.

TABELA DE CUSTOS

Produtos Utilizados No Protótipo	Valores
Placa Arduino Uno	30,00 R\$
Sensor Nível	11,90 R\$
Sensor DHT11	13,00 R\$
Sensor LDR	10 R\$
Capacitor	0,25 R\$
Protoboard	13,00 R\$
Cabos Jumper	15,00 R\$
VALOR TOTAL	93,15 R\$

Esses valores estão baseados nos meus gastos reais podendo variar de site para site e de loja para loja onde no meu caso todas as compras foram feitas no site do MERCADO LIVRE.

CONCLUSÃO

A Internet das Coisas vem crescendo a cada ano e chega com força mostrando que qualquer coisa é possível ser conectada a rede. Apostar nessa tecnologia não só sozinhas mas em conjunto com várias outras faz com que qualquer coisa possa ser criada com muito estudo e calma.

Usar ferramentas, softwares, sensores e placas baratas provaram que é possível fazer algo de baixo custo sendo assim acessível para qualquer pessoa.

Conclui se que a Python em conjunto com o Arduino e seus sensores acompanhado do protocolo MQTT são uma ótima opção para projetos de automação e digitalização de informações e controle das mesmas de baixo custo.

A partir deste projeto, é possível criar não só uma estação meteorológica mas também será possível utilizá-lo na agroindústria e em diversas outras áreas aproveitando o uso de sensores. Transformar esse protótipo um produto real também seria algo interessante futuramente para que possa entrar no mercado.

Esse trabalho contribuirá bastante para a nova leva de pesquisadores que está por vir e que vão buscar conhecimento sobre esse mundo do IoT que ainda é pouco explorado. Com o projeto foi constatado que se pode construir uma estação de baixo custo e que tudo que se imaginava inicialmente foi concretizado.

REFERÊNCIAS

Arduino 2018. Disponível em: <<https://www.arduino.cc/>> Acesso em 23 mar.2018.

Arduino 2018. Disponível em < <https://www.arduino.org/>> Acesso em 23 mar. 2018.

ARDUINO. **What is Arduino.**Arduino.<<http://www.arduino.org/learning/getting-started/what-is-arduino>>. Acesso em: 07 Mar. 2018.

BABU, Suresh. **Aproveitando a Internet of Things(IoT).** IBMdeveloperW orks. <<https://www.ibm.com/developerworks/br/library/iot-key-concepts/>>. Acesso em: 18 j u n . 2018.

BARROS, Marcelo. **MQTT – Protocolos para IoT.**Embarcados. <<https://www.embarcados.com.br/mqtt-protocolos-para-iot/>>. Acesso em: 15 jun. 2018.

BRUNELLI, D.; SARTORI, D.: **A Smart Sensor for Precision Agriculture Powered by Microbial Fuel Cells.** In: Sensors Applications Symposium (SAS), 2016.CARISSIMI, Alexandre. **Internet das Coisas: Middlewares e outras coisas.**

ResearchGate.<https://www.researchgate.net/publication/301298394_internet_das_Coisas_Middlewares_e_outras_coisas>. Acesso em: 07 Ago. 2018.

CLOUDMQTT. **Documentação do CloudMQTT.** Documentation. Disponível em <<https://www.cloudmqtt.com/docs.html>>. Acesso em: 26 jul. 2018.

FISHER, Andrew. **Melhores práticas para desenvolvimento de IoT.** IBMdevelopersW orks.<<https://www.ibm.com/developerworks/br/library/iot-mobile-practices-iot-success/>>. Acesso em: 08 Ago. 2017.

GANTAIT, Amitranjan; MUKHERJEE, Ayan; PRATA, Joy. **Protegendo dispositivos e gateways de IoT.** IBMdeveloperWorks. <<https://www.ibm.com/developerworks/br/library/iot-trs-secure-iot-solutions1/index.html>>. Acesso em: 07 Ago. 2018.

ECLIPSE 2018. Disponível em: <<https://www.eclipse.org/>> Acesso em 09 Ago.2018.

PYTHON 2018. Disponível em <<https://www.python.org/>> Acesso em 08 Ago.2018.

SUHANKO, James. IoT – Configurando um MQTT broker. Do Bit ao Byte. Disponível em .
Acesso em 15 mar. 2018.

MCROBERTS, Michael. Arduino Básico, 1. ed. Tradução de Rafael Zanolli, São Paulo. Novatec, 2011.

FILIFELOP. O que é Arduino. Filipeflop. <<http://blog.filipeflop.com/arduino/o-que-e-arduino.html>>.
Acesso em: 27 jan. 2018.

CLOUDMQTT. Documentação do CloudMQTT. Documentation. Disponível em
<<https://www.cloudmqtt.com/docs.html>>. Acesso em: 26 jul. 2018.

