

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
ESCOLA DE ENGENHARIA / ENGENHARIA ELÉTRICA
Trabalho Final de Curso II

Ygor Nadler da Silva
João Vítor Prado Rodrigues

**PROTÓTIPO PARA MONITORAMENTO DOS PARÂMETROS DE
TEMPERATURA EM MÓDULOS FOTOVOLTAICOS POR INTERNET
DAS COISAS – IoT**

Trabalho Final de Curso II como parte dos
requisitos para obtenção do título de bacharel em
Engenharia Elétrica apresentado à Pontifícia
Universidade Católica de Goiás

BANCA EXAMINADORA:

Prof. MSc. Carlos Alberto Vasconcelos Bezerra – Orientador. ENG-PUC Goiás

Prof. MSc. Carlos Daniel de Souza Bezerra – IFG.

Prof. MSc. Luis Fernando Pagotti – ENG-PUC Goiás.

Goiânia, 09 de junho de 2021.

PROTÓTIPO PARA MONITORAMENTO DOS PARÂMETROS DE TEMPERATURA EM PAINÉIS FOTOVOLTAICOS POR INTERNET DAS COISAS - IoT

YGOR NADLER DA SILVA, JOÃO V. P. RODRIGUES, CARLOS ALBERTO VASCONCELOS BEZERRA

Abstract — *This course conclusion work was committed to evaluating the availability of applying a prototype that would enable the user to monitor, in real time, the temperature parameters of a photovoltaic system, considering the impact that temperature variations have in the production of energy from photovoltaic generators. The objective of the study was to understand the possibility of developing this prototype using the Wi-Fi Module NodeMCU ESP8266 ESP-12E, capable of implementing in any system, the possibility of connecting to the Internet through Wi-Fi.*

The prototype assists in the process of obtaining system temperature data and also enables the implementation of various sensors and actuators, in addition to complex and automatic programs and functions, enabling the prototype user to make intelligent decisions based on their own statistical data of the generator system, since the analysis of the measured results is done in a simple way through the AllThingsTalk Maker application server. Improving the efficiency of these processes through their intelligent automation, using intelligent sensors and Arduino's free programming. So, different tests were carried out, one involving computer simulation of the prototype and the characteristics of a photovoltaic module, and the other that was done through the physical construction of the prototype and measurements carried out at room temperature and in a cup of hot water. It was clear that the prototype was successful in carrying out temperature measurements and making these results available for analysis by the user, which can be accessed over the Internet. It was not possible to develop the measurements on a real photovoltaic module, given the extreme situation in Brazil.

Index Terms— *Internet of Things, Dedicated Networks, Data Transmission, 5G, Termopar MAX 6675, AllThingsTalk, NodeMCU, ESP8266, IoT Devices, Cloud Networking, Temperature in Photovoltaic Modules.*

Resumo — Este trabalho de conclusão de curso se comprometeu em avaliar a disponibilidade de aplicação de um protótipo que capacitasse o usuário a realizar o monitoramento, em tempo real, dos parâmetros de temperatura de um sistema fotovoltaico, tendo em vista o impacto que as variações de temperatura tem na produção de energia dos geradores fotovoltaicos. O objetivo do estudo era compreender a possibilidade de desenvolvimento desse protótipo utilizando o Módulo Wi-Fi NodeMCU ESP8266 ESP-12E, capaz de implementar em qualquer sistema, a possibilidade de conexão a Internet através do Wi-Fi.

O protótipo auxilia no processo da obtenção dos dados de temperatura do sistema e, possibilita ainda, a implementação

de sensores e atuadores diversos, além de programas e funções complexas e automáticas, capacitando o usuário do protótipo a tomar decisões inteligentes, embasadas em dados estatísticos próprios do sistema gerador, uma vez que, a análise dos resultados medidos é feita de forma simples através do servidor de aplicação do AllThingsTalk Maker. Melhorando a eficiência desses processos através da automatização inteligente dos mesmos, utilizando sensores inteligentes e a programação livre do Arduino. De modo que foram feitos testes distintos, um envolvendo a simulação computacional do protótipo e das características de um módulo fotovoltaico e, o outro que foi feito através da construção física do protótipo e de medições realizadas na temperatura ambiente e em um copo de água quente. Restou nítido que o protótipo obteve êxito em realizar as medições de temperatura e disponibilizar esses resultados para análise pelo usuário, podendo ser acessados pela Internet. Não foi possível desenvolver as medições em um módulo fotovoltaico real, tendo em vista a situação extrema do Brasil.

Palavras-chave — Internet das Coisas, Redes Dedicadas, Transmissão de Dados, 5G, Termopar MAX 6675, AllThingsTalk, NodeMCU, ESP8266, Dispositivos IoT, Rede na Nuvem, Conectividade, Temperatura em Módulos Fotovoltaicos.

I. INTRODUÇÃO

A interconectividade entre dispositivos permitiu que o usuário ou observador se abstevesse fisicamente dos processos envolvendo dispositivos eletrônicos, sendo responsável somente por implementar o sistema e monitorá-lo. O monitoramento desses processos é realizado através da análise dos dados disponibilizados pelos sensores inteligentes, que estão conectados entre si e, com a Nuvem. Essa conexão com a Nuvem, que é nada mais que o nome comumente utilizado para se referir a um servidor online que está sempre disponível, permite o acesso aos dados captados pelos sensores em qualquer lugar do mundo, desde que o usuário possua Internet e consiga se conectar ao servidor (Nuvem) onde os dados estão sendo armazenados.

Essa facilidade na interação entre o usuário, que pode constantemente estar monitorando seu sistema, aprimorando a eficiência, buscando erros e realizando diagnósticos, e o sistema, que em tempo real, apresenta resultados e realiza análises próprias que auxiliam na manutenção, diminuem o tempo em que processos comuns acontecem e, podem aumentar consideravelmente a eficiência e, até mesmo, o tempo de vida de sistemas e equipamentos eletroeletrônicos como, transformadores, módulos fotovoltaicos, capacitores,

inversores de frequência e entre outros, é o ponto principal que motiva o desenvolvimento deste estudo. Uma vez que, para a realidade dos geradores fotovoltaicos, que tem sua produção afetada drasticamente pelo aumento da temperatura nos módulos, é necessário o desenvolvimento de sistemas capazes de analisar a realidade desses equipamentos, possibilitando o monitoramento dessas grandezas e, permitindo interferências pontuais quando necessárias.

O conceito de Internet das Coisas passa a ocupar várias esferas do cotidiano humano, sendo realizados diversos estudos e testes para a aplicação desses, ou novos, padrões de tecnologia sem fio, buscando o aprimoramento na eficiência dos processos. Podendo ser aplicada até na alimentação energética dos dispositivos, como o caso da Coreia do Sul que, em 2013, inaugurou a primeira estrada eletrificada capaz de alimentar os veículos elétricos através da ressonância magnética gerada pelos cabos elétricos enterrados abaixo do asfalto [1]. Esse é mais um dos exemplos das inúmeras aplicações da tecnologia sem fio.

Este trabalho apresenta como proposta a utilização do Módulo Wi-Fi NodeMCU ESP-12E, que é uma placa de desenvolvimento que combina o chip ESP8266 com a pilha de protocolo TCP/IP integrada, permitindo que o usuário implemente o acesso a rede Wi-Fi com qualquer microcontrolador, inclusive da linha Arduino. O sistema tem o intuito de apresentar a conectividade de plataformas de IoT, em Nuvem, para aplicações em instrumentação industrial. Especificamente, para a transmissão dos dados de temperatura dos módulos fotovoltaicos, medidos pelo Termopar Max 6675 e monitorados pela Internet, através do aplicativo do AllThingsTalk Maker, sem necessidade de estar presente no local da instalação. Além disso, este estudo trata também da caracterização da Internet das Coisas, conceito importantíssimo que, de forma simplificada, abrange a conexão e troca de informação constante entre dispositivos, permitindo a automação de processos cotidianos e, o monitoramento em tempo real de eletrônicos, comunidades, grandezas físicas e climáticas, entre outros. Avaliando a disponibilidade da construção desse protótipo e as possibilidades que surgem com a tecnologia IoT em conjunto com os servidores em Nuvem.

Nas seções seguintes, serão detalhadas as etapas de construção do circuito eletrônico, além do estabelecimento dos conceitos primordiais para o entendimento do Módulo ESP8266, descrevendo: a metodologia utilizada para a montagem e testagem do sistema e, por fim, são descritos os resultados e as conclusões gerais sobre o trabalho.

II. FUNDAMENTAÇÃO TEÓRICA

A comunicação é um fator indispensável para o convívio em sociedade e o entendimento da desenvoltura da tecnologia móvel é extremamente importante para a compreensão da mudança nos processos de comunicação entre seres vivos e dispositivos. A facilidade na comunicação estimulou as pessoas a fazerem uso dessas novas tecnologias que transformaram as

interações sociais, permitindo o compartilhamento de mensagens, audios, vídeos e outros tipos de multimídia.

Atualmente, existe um horizonte totalmente revolucionário ao alcance da sociedade, que é definido pela Quinta Geração (5G) de telefonia móvel, onde estar conectado e atualizado será simples como “respirar”.

A. Tecnologia sem fio

Uma forma de se classificar os sistemas sem fios é avaliando o tipo de ligação necessária entre os elementos que compõe o modelo. Para um modelo ponto-a-ponto, o objetivo é a interligação de dois pontos sem a necessidade de atender requisitos de alteração de local como é o caso de sistemas com mobilidade. Como os descritos na Figura 1 abaixo.

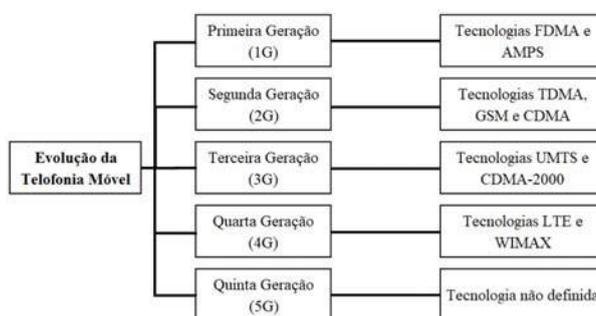


Figura 1 - Diagrama da evolução da telefonia móvel. (Fonte: Do 1G ao 5G: Evolução das redes de telefonia móvel)

Já para um modelo ponto multi-ponto, a intenção é cobrir uma certa área de atuação [2]. As redes sem fio surgiram como redes complementares às redes cabeadas, com intuito de promover a mobilidade e a visualização rápida dos dados independentes da localização do usuário. Elas foram padronizadas segundo o IEEE, que regulamentou a norma 802.11 e as suas versões dividindo em quatro grandes grupos, definidos pelo seu raio de alcance [3].



Figura 2 - Representação dos padrões globais de rede sem fio e sua área de cobertura. (Fonte: Redes Sem Fio: Tecnologias de Redes Sem Fio)

Os sistemas WPAN (*Wireless Personal Area Network*) são caracterizados por uma pequena área de cobertura, com raio

aproximado de algumas dezenas de metros e, uma conexão direta entre dispositivos, sem a necessidade de um elemento central para conexão à rede cabeada. São utilizados para conexão de dispositivos que estejam fisicamente próximos e que podem se conectar à rede cabeada e servirem de ponto de conexão. Comumente aplicada para conexão de dispositivos como impressoras, PDA (*Personal Digital Assistents*), celulares e periféricos diversos.

Para esse padrão, a tecnologia de comunicação mais conhecida mundialmente é o Bluetooth (IEEE 802.15.1). Em comparação com outros padrões de rede sem fio, este possui baixa taxa de transferência de dados, ficando abaixo dos 600 quilobits por segundo, e baixo alcance de cobertura [2]. Porém, o mesmo apresenta um baixo consumo de energia, sendo uma ótima opção para a conexão entre pequenos dispositivos.

Segundo PEREIRA (2013), a diferença básica entre uma PAN e uma LAN doméstica é que a primeira é centrada em uma pessoa, para a comunicação de um computador com outros dispositivos. Já a LAN é uma conexão local e o objetivo é que ela possa ser utilizada por vários usuários [4].

Os sistemas WLAN (*Wireless Local Area Network*) são redes locais com alcance limitado a um raio de 100 a 300 metros, comumente usadas em escritórios, shoppings centers, residências e instituições de ensino. Para este padrão, a tecnologia mais difundida mundialmente é o Wi-Fi (*Wireless Fidelity*) que permite operação em quatro taxas de transmissão na camada física, estabelecendo regiões de cobertura, para cada taxa, ao redor do ponto de acesso (AP – *Access Point*) [5].

Segundo KIOSKEA (2009), a grande implementação das redes locais em espaços públicos comerciais trouxe a diminuição dos custos com a aquisição dos equipamentos e, conseqüentemente, levou a disseminação de redes particulares que hoje em dia, são extremamente comuns [3].

O Wi-Fi é um conjunto de especificações para redes locais sem fio baseado no padrão IEEE 802.11, que foi desenvolvido pelo grupo WECA (*Wireless Ethernet Compatibility Alliance*). O grupo nasceu em 1999 com a união de algumas empresas como 3com, Nokia, Lucent Technologies (atualmente Alcatel-Lucent) e Symbol Technologies (adquirida pela Motorola) e, passou a se chamar Wi-Fi Alliance em 2003. As especificações IEEE 802.11 foram desenvolvidas em cima das especificações IEEE 802.3, que era conhecido pelo nome Ethernet e simplesmente consiste na grande maioria das tradicionais redes sem fio.

O fato de ambas especificações serem muito semelhantes se mostrou uma vantagem significativa para a implementação do Wi-Fi, já que não foi necessário a criação de nenhum protocolo específico para a comunicação de redes sem fios baseadas nessa nova tecnologia [5]. A tecnologia é caracterizada pela transmissão de dados por sinais de radiofrequência, que cobriam áreas de até centenas de metros. Sua primeira versão foi lançada em 1997, após aproximados 7 anos de estudo, ficando conhecida atualmente como 802.11-1997 ou, ainda, como 802.11 *legacy*. Por se tratar de uma tecnologia de transmissão por radiofrequência, o IEEE determinou que o padrão operasse no intervalo de frequências entre 2,4 GHz e 2,4835 GHz. Esse intervalo de frequência representa uma das faixas ISM (*Industrial, Scientific and*

Medical) que são intervalos de frequência disponíveis para o uso sem a necessidade de aprovação direta das entidades responsáveis que, no caso do Brasil, é a ANATEL (Agência Nacional de Telecomunicações).

As faixas ISM são bandas reservadas internacionalmente para o desenvolvimento industrial, científico e médico, compreendendo os seguintes intervalos de frequência: 902 MHz – 928 MHz; 2,4 GHz – 2,485 GHz e 5,15 GHz – 5,825 GHz. Essas tecnologias utilizam as bandas ISM devido as poucas restrições existentes para o uso dessas faixas de frequência. As únicas restrições estão relacionadas a potência máxima transmitida e com a largura de banda que deve ser mantida dentro dos limites [6].

Os sistemas WMAN (*Wireless Metropolitan Area Network*) atendem áreas metropolitanas com raio de mais de um quilômetro. Também conhecidas pelo nome de Ane local rádio (LBR), as WMANs são normatizadas pela IEEE 802.16, onde suas redes mais antigas atingem taxas de transferências de dados de 1 a 10 Mbit/s, com alcance de 4 a 10 quilômetros [3].

Seu funcionamento é similar ao da tecnologia do Wi-Fi, diferenciando pela maior área de atuação, as velocidades superiores e o maior número de usuários. Dentre as características da conexão WIMAX, suas grandes vantagens são: alto desempenho, com antenas inteligentes e menor custo de infraestrutura, criação de uma rede de cobertura de conexão com a internet parecida com a de cobertura de celular, tornando o acesso à internet de qualidade mesmo em movimento, oferta em regiões sem infraestrutura para cabearamentos e independência das redes de concessionária de telefonia [7].

Além disso, a tecnologia opera tanto em faixas de frequência licenciadas quanto as isentas de licença. Isso permite um uso mais amplo do padrão IEEE 802.16 e garante que o mesmo possa aproveitar da maior potência disponível para downlink oferecido pelas faixas licenciadas.

Faixa	Frequências	Requer Licença?	Disponibilidade
2,5 GHz	2,5 a 2,69 GHz	Sim	Alocada no Brasil, México, alguns países do sudeste asiático e nos EUA (o WiMax Fórum também inclui 2,3 GHz nesta categoria de faixa, pois "se espera que o [2,3 GHz] seja coberto pelo rádio de 2,5 GHz".)
3,5 GHz	3,3 a 3,8 GHz, porém principalmente de 3,4 a 3,6 GHz	Sim, em alguns países	Na maioria dos países, a faixa de 3,4 GHz a 3,6 GHz é alocada para o wireless de banda larga.
5 GHz	5,25 a 5,85 GHz	Não	Na porção de 5,725 GHz a 5,85 GHz, muitos países permitem uma potência maior, o que pode melhorar a cobertura.

Figura 3 - Quadro das faixas de frequências disponíveis para o WIMAX. (Fonte: Do 1G ao 5G: Evolução das redes de telefonia móvel)

Os sistemas WWAN (*Wireless Wide Area Network*) são utilizados para atender áreas ainda maiores, motivo que é conhecida também como Rede Continental, já que sua área de cobertura se estende por um país ou, até mesmo, por um continente inteiro. O objetivo de uma área de cobertura tão

extensa é proporcionar a conexão entre cidades ou diferentes MANs, garantindo uma conexão rápida e estável até para usuários ou dispositivos se movendo em alta velocidade [3].

Em março de 2002, formou-se o grupo MBWA (*Mobile Broadband Wireless Access*), com intuito de avaliar o interesse e a viabilidade da implementação de um novo padrão de conexão sem fio. O estudo concluiu que existiam necessidades diferentes a serem atendidas, motivando a criação de um grupo específico para o desenvolvimento do novo padrão IEEE 802.20. Conhecido como Mobile-Fi, o novo padrão surge para transformar o conceito de rede sem fio, prometendo a mesma área de cobertura dos sistemas de telefonia móvel mas com a velocidade e estabilidade das conexões Wi-Fi [8].

A Figura 4 abaixo, faz um comparativo entre os padrões IEEE 802, indicando as tecnologias de acesso ao meio principais de cada padrão e confrontando os resultados de cada padrão quanto aos critérios de: mobilidade, largura de banda utilizada, taxa de dados de pico por usuário, bem como a realidade do padrão, se já foi implantada comercialmente (comercial) ou se ainda está em desenvolvimento/padronização (proposta). Atualmente, os padrões IEEE 802.16a, 802.16e e 802.15.3a já foram implantados, estando disponíveis para a utilização pelos usuários.

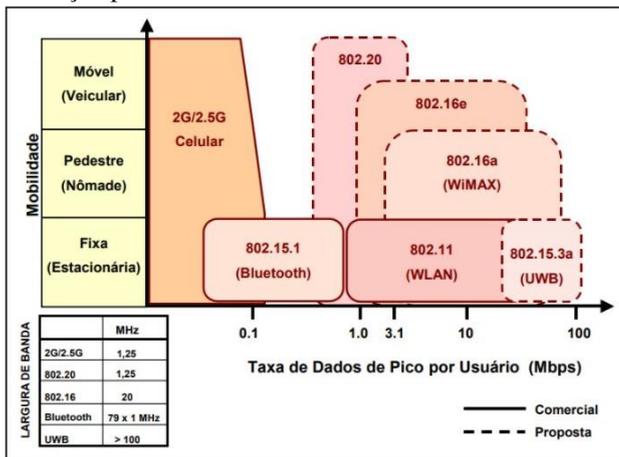


Figura 4 - Comparativo entre os padrões de rede sem fio e suas respectivas tecnologias de acesso ao meio. (Fonte: LIMA; SOARES; ENDLER (2004))

A Quinta Geração (5G) de telefonia móvel representa o acumulado dos avanços na área da comunicação sem fio e dos desenvolvimentos tecnológicos mais recentes. Segundo o presidente da Agência Nacional de Telecomunicações (Anatel), Leonardo Euler de Moraes, em entrevista ao programa Brasil em Pauta na TV Brasil, “Não se trata de mais um G e sim um guarda-chuva que envolve e potencializa outras tecnologias”. Completa que, essa tecnologia será catalisadora de outras tecnologias como a Internet das Coisas – IoT, inteligência artificial, robótica, inteligência mista e aumentada, entre outras [9]. Ainda no primeiro semestre desse ano de 2021 acontecerá o Leilão do 5G, com edital em análise pelo Tribunal de Contas da União (TCU), onde será comercializado o direito para uso da faixa de frequência de transmissão do 5G, que é uma das etapas necessárias para a implementação completa da rede,

onde será realizado licitações das radiofrequências nas faixas de 700 MHz, 2,3 GHz, 3,5 GHz (separada para a conexão sem fio de banda larga, conforme Figura 5) e 26 GHz [10].



Figura 5 – Diagrama ilustrativo do desenvolvimento dos padrões de tecnologia móvel até o 5G. (Fonte: https://www.teleco.com.br/5g_tecnologia.asp)

A tecnologia 5G, conforme Figura 5 acima, compreende as diversas tecnologias sem fio que habitam os padrões sem fio descritos anteriormente. Proporcionando conexões mais rápidas, mais estáveis, com menor latência e com uma capacidade muito maior de usuários. O que permitirá, por exemplo, velocidades de até 10 Gbps (gigabits por segundo, equivalente a 30 vezes o entregue pelo 4G), latência de 1 ms (milissegundos) e densidade de conexão de até 1 milhão/km² (um milhão por quilômetro quadrado, equivalente a 100 vezes o entregue pelo 4G) [11]. O padrão do 5G teve suas especificações estabelecidas pelo *release 15*, concluído pela 3GPP (*3rd Generation Partnership Project*) ainda em 2017, e vem sendo atualizado desde então. A 3GPP é um grupo que une sete organizações de desenvolvimento de padrões de telecomunicações, conhecidas como “Parceiros Organizacionais” e, fornece a seus membros um ambiente estável para produzir os “Relatórios e Especificações” que definem as tecnologias 3GPP [12].

A implementação desse novo padrão, terá diversas implicações tecnológicas e socioeconômicas para o Brasil e para o mundo. “*Só em questão de geração de emprego em tecnologia da informação, inteligência artificial, estamos prevendo mais de um milhão de novos trabalhadores qualificados entrando no setor, porque o 5G vai permear em todas as áreas da economia brasileira. É na saúde, na educação, na mídia, em tudo. Vai ser um mundo de dados trafegando com muito mais velocidade e menor latência*”, afirma Vivien Suruagy, presidente da Federação Nacional de Instalação e Manutenção de Infraestrutura de Redes de Telecomunicações e de Informática (Feninfra), que brinca que o 5G tem capacidade para transformar o mundo em um episódio de “Os Jetsons” [10].

Porém, essa evolução nos parâmetros de velocidade, latência e densidade de conexão não são os maiores atrativos que destacam a tecnologia 5G, mas sim, a Virtualização de Funções de Rede (*Network Function Virtualisation, NFV*). Essa tecnologia é considerada um dos principais componentes que compõe a evolução da arquitetura da rede sem fio para prover serviços flexíveis, organizados em fatias e de baixo custo.

Em um estudo desenvolvido em 2020, no Departamento de Ciência da Computação da Universidade de Brasília, foi projetado, implementado e avaliado uma infraestrutura experimental de rede móvel virtualizada, onde foi avaliado o comportamento do ambiente virtualizado e mostrado o potencial da integração da proposta à arquitetura 5G como

elemento de suporte à otimização da infraestrutura virtualizada [13]. O NFV no 5G propõe a abstração da infraestrutura de recursos para permitir o uso de elementos lógicos de rede na implementação e implantar funções de rede em dispositivos de borda, promovendo um aumento significativo no tráfego de informações na rede.

B. Protótipo

O protótipo construído neste estudo utiliza o circuito ESP 8266-12E, que é um microcontrolador com capacidade de conexão Wi-Fi, não necessitando de nenhum módulo externo para conectar com redes Wireless, conforme estabelecido anteriormente no detalhamento das tecnologias sem fio. A programação se dá através de comandos AT e pode ser desenvolvida utilizando a interface de programação do Arduino (IDE). O Arduino é necessário para aumentar a capacidade de entradas e saídas do sistema. A comunicação com o Arduino se estabelece através da comunicação serial RX, TX do Hardware de ambos. Para melhor compreender os processos envolvidos nessa relação descrita acima, é necessário o detalhamento de seus componentes.

O módulo MAX6675 Termopar Tipo K – 0° a 800°C é um dispositivo eletrônico que tem como finalidade fazer a medição de temperatura, sendo um conversor serial de resolução de 12-bits que realiza compensação de junção fria, correção de linearidade, detecção de defeito no termopar. A sonda que faz parte do sensor é revestida em aço inoxidável e a ponta possui blindagem. Esta forma de construção do termopar permite que o mesmo possa ser utilizado em temperaturas mais altas. Pode ser utilizado em aplicações industriais, em aparelhos eletrônicos ou em projetos pessoais.

Sua composição é feita a partir de dois metais distintos que são unidos e ligados a um dispositivo que tenha capacidade de ler termopares e, se há uma diferença de temperatura entre a extremidade unida e as extremidades livres, uma diferença de potencial surge e se o termopar estiver conectado ao dispositivo que tenha capacidade de interpretar o sinal, um valor de temperatura correspondente será apresentado.

Após ter feito a medição da temperatura do ambiente/local, será utilizado o protocolo SPI (*Serial Peripheral Interface*) de três fios, uma tecnologia de protocolo de comunicação serial de dados realizado com dispositivos periféricos, de forma rápida e em tempo real.



Figura 6 – Sensor de Temperatura MAX6675 Termopar

O Arduino foi criado em 2005 por um grupo de 5 pesquisadores: Massimo Banzi, David Cuartielles, Tom Igoe,

Gianluca Martino e David Mellis. O objetivo era elaborar um dispositivo que fosse ao mesmo tempo barato, funcional e fácil de programar, sendo dessa forma acessível a estudantes e projetistas amadores. Além disso, foi adotado o conceito de hardware livre, o que significa que qualquer um pode montar, modificar, melhorar e personalizar o Arduino, partindo do mesmo hardware básico [14].

É um microcontrolador que adota a ideia de hardware livre, o que significa que qualquer um pode montar, modificar, melhorar e personalizar o Arduino, partindo do mesmo hardware básico. Possui pinos digitais que podem ser programados como entradas ou saídas. A conexão com sensores analógicos pode ser realizada pelas entradas analógicas, que são conversores A/D de 10 bits. A linguagem de programação é derivada do C/C++ e a programação é realizada pela IDE (Integrated Development, ou Ambiente de Desenvolvimento Integrado). Existem diversas possibilidades de utilização, vários sensores e componentes compatíveis.



Figura 7 – Sensor de Temperatura MAX6675 Termopar Tipo K

Depois de programado, o microcontrolador pode ser usado de forma independente, existindo diversas possíveis aplicações e possibilidades de utilização da tecnologia. Para este trabalho, será utilizado sensores de tensão, corrente e temperatura. Na programação do Arduino, existem alguns comandos mais elaborados, onde sua utilização necessita da inclusão de uma biblioteca na linha de programação. Uma biblioteca é definida como um conjunto de instruções desenvolvidas para executar tarefas específicas relacionadas a um determinado dispositivo [15]. Várias bibliotecas da Adafruit e Sparkfun, podem ser acessadas pelo GitHub e utilizadas para programação da plataforma Arduino.

Na proposta de trabalho a ser desenvolvido, as seguintes bibliotecas serão utilizadas, com a finalidade de estabelecer os programas para o Arduino e o NODE MCU 8266-12E: LiquidCrystal.h; max6675.h; ESP8266WiFi.h; ESP8266WebServer.h [16]. Estas bibliotecas facilitam o processo de programação, pois são específicas para cada situação encontrada no ambiente de aplicação dos dispositivos eletrônicos envolvidos no trabalho.

Onde, a biblioteca LiquidCrystal.h permite o controle de display LCD baseado no chip de controle HD44780 ou compatível, que é encontrado na maioria dos LCDs baseados

em texto. A biblioteca funciona no modo de 4 ou 8 bits (ou seja, usando 4 ou 8 linhas de dados além das linhas de controle rs, enable, opcionalmente as linhas de controle RW). Possuindo várias funções para o controle da escrita em LCDs.

A biblioteca Max6675.h realiza a leitura das informações geradas no circuito integrado (CI). A biblioteca SPI.h é utilizada para estabelecer o protocolo de comunicação SPI entre os circuitos integrados. Além delas, a biblioteca SoftwareSerial foi desenvolvida para permitir a comunicação serial em outros pinos digitais do Arduino, utilizando software para replicar a funcionalidade. Com isso, é possível ter várias portas seriais de software, com velocidades de até 115200 bps (bits por segundo).

Por fim, incluiu-se a biblioteca AllThingsTalk_WiFi.h, que facilita a conexão dos dispositivos com o aplicativo AllThingsTalk Maker, estabelecendo a conexão com a Nuvem de forma prática e descomplicada. Para este estudo, foram considerados os comandos já fornecidos pelo servidor de aplicação AllThingsTalk. Que estabelece a interconectividade entre nosso ativo e nosso servidor de aplicação, através da placa ESP8266. Maiores detalhes quanto a implementação desses comandos e dos processos envolvendo a utilização do AllThingsTalk Maker, se encontram nos Apêndices.

O módulo Wifi ESP8266 NodeMCU ESP-12E é um chip compacto projetado para atender as necessidades de conectividade sem fio (padrão 802.11 B/G/N) podendo tanto ser utilizado independentemente ou em conjunto com um microcontrolador externo (Arduino, PIC, ARM, etc). Com este módulo é possível adicionar conectividade Wi-Fi (802.11) a qualquer sistema. Modos de operação AP e station (Access Point ou Estação). Possui protocolos da camada de rede internos (Stacks), não necessitando de biblioteca adicional ou firmware atualizável. A programação deste pode ser realizada através de comandos AT e utilizando a IDE do Arduino.

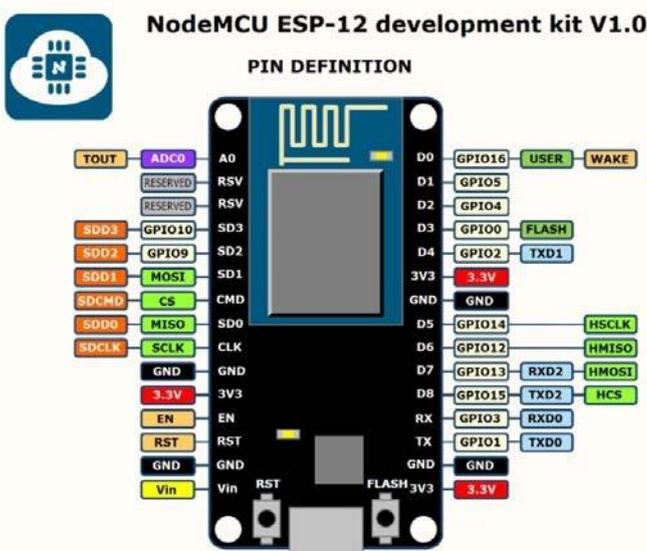


Figura 8 – Diagrama ilustrativo do módulo Wi-Fi ESP8266 NodeMCU ESP-12E (Imagem: <https://www.filipeflop.com/blog/esp8266-nodemcu-como-programar/>)

O módulo tem dois botões, conforme mostrado na imagem abaixo: Flash (utilizado na gravação do firmware) e RST (Reset). No mesmo lado temos o conector micro usb para alimentação e conexão com o computador. No lado oposto, temos o ESP-12E e sua antena embutida, já soldado na placa. Nas laterais temos os pinos de GPIO, alimentação externa, comunicação [17]. Conforme a estrutura demonstrada na Figura 8 acima, contendo a representação dos pinos do módulo em questão.

Também amplamente chamados por sua tradução em inglês, os Application Servers, assim como o nome sugere, servem para que aplicações sejam acessadas por clientes via rede. A plataforma de um servidor de aplicação contém todos os módulos necessários para que determinado código de uma aplicabilidade seja executada e cumpra seu objetivo, respondendo aos clientes que acessam essas aplicações e lhes entregando as informações requisitadas.

O objetivo de um servidor de aplicação é facilitar o acesso às aplicações sem que seja necessário instalar seus módulos nos clientes [18]. Assim, o gerenciamento fica centralizado e se torna mais fácil mantê-lo. Essas facilidades de gerenciamento centralizado não beneficiam apenas os clientes que acessam a aplicação, mas também agilizam o trabalho dos desenvolvedores e demais atividades de deploy, que é uma fase do ciclo de vida de um software.

O funcionamento do servidor de aplicação vai depender da linguagem de programação, das necessidades de interconectividade com redes e bancos de dados, bibliotecas requisitadas pela aplicabilidade em si. Em ambientes legados, um servidor de aplicação deve aparecer como programa que estritamente realiza o trabalho de executar o código-fonte de uma aplicabilidade e a servir de fato aos clientes.

Também é comum encontrar o uso de Terminal Services dentro desse contexto, para manter um aplicativo que originalmente seria instalado no cliente, dentro de um servidor e proporcionar um acesso remoto. Isso ocorre tanto por facilidade de gerenciamento quanto por razões de economia com licenças.

Já no contexto de cloud, você encontrará servidores de aplicação em uma abordagem mais de Platform as a Service. Isso porque um servidor de aplicação na nuvem já terá, na sua configuração inicial, todos os requisitos necessários para que uma aplicabilidade seja servida, restando apenas fazer o deploy de seu código-fonte [18].

Tratando de cloud, não apenas a parte de instalação e suas boas práticas já são entregues de forma simples e prática, mas também a manutenção da segurança, backup, restore e tantas outras coisas que um servidor de aplicação demanda à uma equipe de TI. Tudo isso em um painel objetivo e simplificado e, muitas vezes, de forma automática.

Nesse estudo, o servidor de aplicação utilizado é o AllThingsTalk, que detém a plataforma do AllThingsTalk Maker, que é uma plataforma de desenvolvimento da AllThingsTalk para conectar rapidamente seus dispositivos e ativos, permitindo a coleta e análise, em tempo real, dos dados obtidos. AllThingsTalk Maker é gratuito para fins de prototipagem e, o AllThingsTalk, disponibilizou todo o passo a passo dos processos para estabelecer a conexão com entre o

sistema desenvolvido neste estudo e a nuvem [19]. Permitindo o monitoramento do sistema por parte do usuário.

Existem ainda, outros servidores de aplicação que disponibilizam funções similares, e até mesmo, funções mais abrangentes, como o Google Cloud, que foi desenvolvido com o objetivo de favorecer a execução de uma série de serviços com base na computação em nuvem, permitindo que empreendimentos contem com uma alta performance, segurança e confiabilidade em seus processos, otimizando a qualidade dos resultados adquiridos [20], ou o Microsoft Azure, que é uma plataforma de nuvem com mais de 200 produtos e serviços de nuvem projetados para ajudar na implementação de novas soluções que otimizem o solucionamento de problemas, permitindo a criação, execução e o gerenciamento de aplicativos em várias nuvens, locais e na borda [21]. Porém, a plataforma do AllThingsTalk, utilizada no estudo, dispunha do esclarecimento necessário para que fosse executado o protótipo pretendido.

A computação em nuvem, cloud computing, se tornou cada dia mais fundamental para impulsionar os negócios e agilizar o trabalho das equipes. Grandes companhias mundiais utilizam essa tecnologia de armazenamento. Ao substituir o armazenamento de dados em desktop para o armazenamento na nuvem, é possível garantir a segurança das informações e ainda ampliar as formas de acesso a elas. Por isso, a nuvem tem ganhado relevância no presente e deve manter sua importância no futuro [22].

Não é necessário realizar grandes investimentos iniciais em hardware para a empresa, e perder tempo nas atividades de manutenção e gerenciamento do mesmo. Existem outras vantagens de grande importância como, agilidade e praticidade, pois esse serviço oferece acesso rápido a recursos de TI flexíveis e de baixo custo [23]. Essa é uma das primeiras questões que deve passar na sua mente ou nas de seus diretores quando o assunto é um novo investimento. A adoção de serviços cloud elimina gastos de capital com: a compra de hardware e alguns software, e instalação e execução de datacenters locais. Isso tudo inclui a economia com racks de servidores, disponibilidade constante de eletricidade para energia e resfriamento.

A maior parte dos serviços de computação em nuvem é fornecida sob demanda, com o objetivo de que mesmo grandes quantidades de recursos possam ser provisionadas em questão de instantes. Os serviços mais básicos de computação em nuvem, normalmente são adquiridos com apenas alguns cliques. Isso significa que o tempo necessário para disponibilizar os recursos aos desenvolvedores é reduzido de semanas para apenas minutos. O que resulta em um aumento dramático na agilidade da organização, pois, não apenas o custo, mas o tempo necessário para desenvolvimento é substancialmente mais baixo [23].

Um dos mais incríveis benefícios proporcionados pela computação em nuvem é a capacidade de dimensionamento elástico. Isto é, capacidade em fornecer a quantidade correta de recursos de TI necessários. Para esclarecer um pouco mais, a potência de computação, o armazenamento e a largura de banda

podem aumentar ou diminuir conforme seja necessário, não importando a origem geográfica. Os datacenters locais exigem inúmeros equipamentos e implementações, como configuração de hardware, correção de software e outras diversas tarefas de gerenciamento da TI que consomem seu tempo e produtividade. A computação em nuvem é capaz de remover a necessidade de muitas dessas tarefas.

Os maiores serviços de computação em nuvem são executados em uma rede mundial de datacenters seguros, os quais são atualizados regularmente com hardwares de grande velocidade e desempenho, são de última geração. E diversos são os benefícios obtidos através de um único datacenter corporativo oriundo da computação em nuvem, incluindo latência de rede reduzida para aplicativos e mais economia de escalonamento. Isso tudo faz com que seja eliminado as suposições ao determinar sua necessidade de capacidade de infraestrutura. Com a computação em nuvem, os problemas quanto a lidar com a ociosidade de recursos caros ou com limites de capacidade não existem mais. Muitos provedores de computação em nuvem oferecem um amplo conjunto de políticas, tecnologias e controles que fortalecem sua postura geral de segurança. E tudo isso ajuda a proteger os dados, os aplicativos e a infraestrutura contra possíveis ameaças [23].

III. METODOLOGIA

Com o intuito de monitorar variáveis temperaturas em um sistema fotovoltaico, utilizando tecnologia de IoT, foi desenvolvido um protótipo com a capacidade de conexão a internet através da rede Wi-Fi e os dados transmitidos para nuvem utilizando o aplicativo AllThingsTalk, para posterior uso dos mesmos. O projeto foi dividido em duas etapas. Na primeira etapa foi realizado as simulações computacionais para ajustes práticos dos sistemas envolvidos. Na segunda etapa foi desenvolvido o Hardware e os softwares necessários aos sistemas.

A. *Simulação Computacional do sistema*

A simulação de partes do sistema proposto foi realizada no Software ISIS PROTEUS, do fabricante Labcenter (<https://www.labcenter.com/downloads/>). Este software não dá suporte para realização da simulação completa do trabalho proposto. A Figura 9 abaixo, ilustra os pinos de conexões utilizados no Arduino UNO que foi utilizada na simulação. A conexão do CI do sensor de temperatura (MAX 6675) com o Arduino é realizada de forma digital, utilizando os pinos 4, 5 e 6 (SCK, CS e SO, respectivamente). O protocolo utilizado para esta comunicação é o SPI.

As conexões do display LCD foram realizadas através dos pinos 7, 8, 9, 10, 11 e 12 nas sequências indicadas (RS, E, D4, D5, D6 e D7). A Figura 10 abaixo, ilustra também as conexões e os pinos utilizados no display LCD.

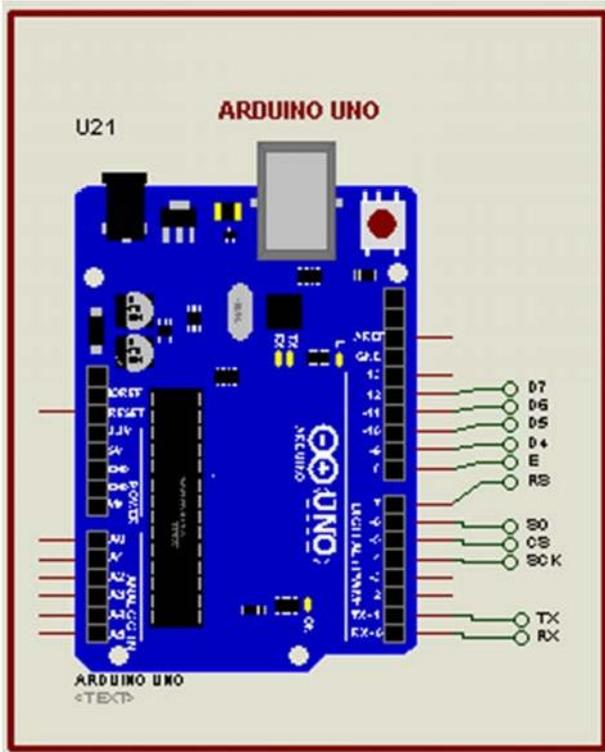


Figura 9 – Simulação: Conexões no Arduino

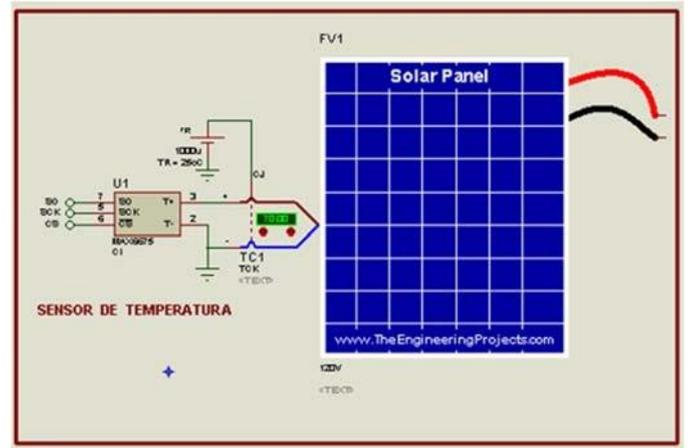


Figura 11 – Simulação: Sensor de temperatura

A saída do sensor de temperatura é um sinal digital com protocolo SPI, que é tratado pela função `readThermocouple()` na programação do Arduino (Anexo 1). A Figura 12 abaixo, apresenta a variável medida no painel Fotovoltaico, através do terminal virtual do Proteus, para uma temperatura de, aproximados, 70 °C no módulo.

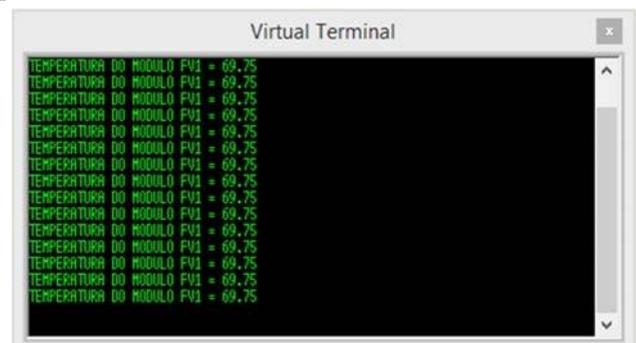


Figura 12 – Simulação: Variáveis medidas no trafo

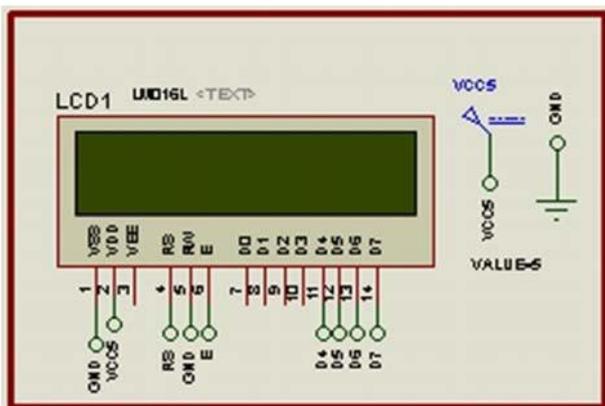


Figura 10 – Simulação: Conexões no display LCD

A Figura 11 abaixo, ilustra o esquema elétrico do sensor de temperatura que utiliza o Circuito Integrado (CI) MAX 6675. Este CI digitaliza as informações de temperatura realizadas com auxílio de um termopar tipo K (TCK), fazendo a correção da junta fria, e transmite esta informação através do protocolo SPI para o Arduino. A função `readThermocouple()` foi utilizada para realizar a leitura do sensor (Apêndice 2), pois a biblioteca `max6675.h`, que poderia ser utilizada, possui falhas com o uso no Proteus. O sistema apresentado foi utilizado para simulação da variável temperatura em painéis Fotovoltaicos com a finalidade de ajustes dos sensores para as medições e posterior uso destes no protótipo final. A programação do Arduino foi realizada com auxílio da biblioteca, `LiquidCrystal.h` e o programa final para simulação está indicado no Apêndice 2.

Na Figura 13 abaixo, tem-se esta grandeza física indicada no Display LCD do sistema. Esta variável será enviada ao NODE MCU 8266 para transmissão a nuvem e posterior uso da mesma no processo de monitoração das temperaturas dos painéis Fotovoltaicos.

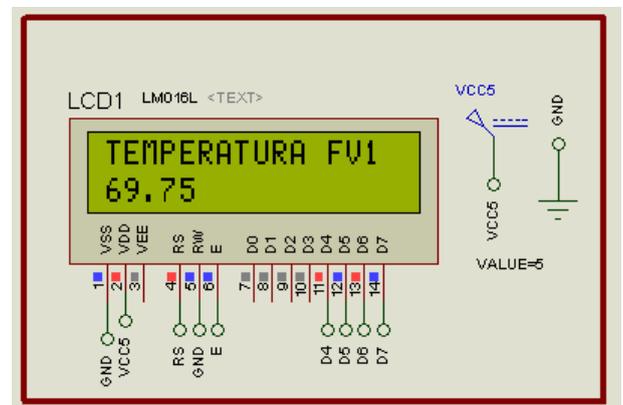


Figura 13 – Simulação: Temperatura medida indicada no LCD

B. Montagem do protótipo

O Hardware do sistema foi montado em bancada para o estudo proposto no trabalho. A Figura 14 abaixo, apresenta o diagrama de blocos do sistema proposto.

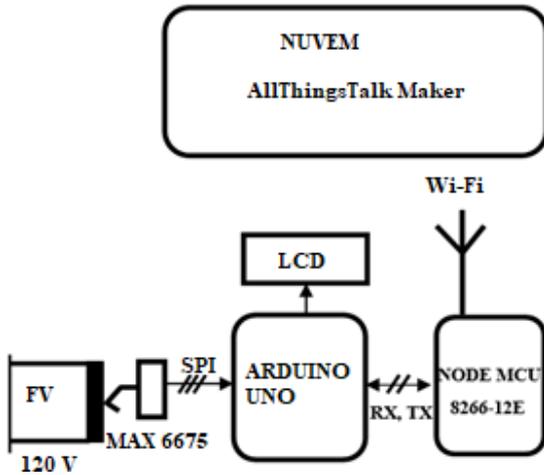


Figura 14 – Diagrama de blocos do protótipo

Neste diagrama é possível identificar o sensor de temperatura composto pelo CI MAX 6675 e o termopar Tipo K. O sensor de temperatura comunica com o Arduino pelas portas digitais (D4, D5 e D6) através do protocolo SPI. A comunicação do Arduino com o NODE MCU 8266-12E é realizado por portas seriais, que foram estabelecidas pela biblioteca SoftwareSerial.h, já que as portas nativas de ambos sistemas foram utilizadas para gravação e depuração. Como o Arduino trabalha com uma tensão de 5V e o NODE MCU com 3,3V foi colocado um conversor digital entre as portas RX e TX de ambos, com a finalidade de adequar os níveis de tensões. O sensor de temperatura MAX6675 foi adquirido montado em placa de circuito impresso e foi conectado ao sistema de modo que a temperatura pudesse ser monitorada e este conectado diretamente ao Arduino.

A Figura 15 abaixo, apresenta uma fotografia do Hardware desenvolvido para o estudo proposto. Nesta figura é possível verificar cada componente do sistema.

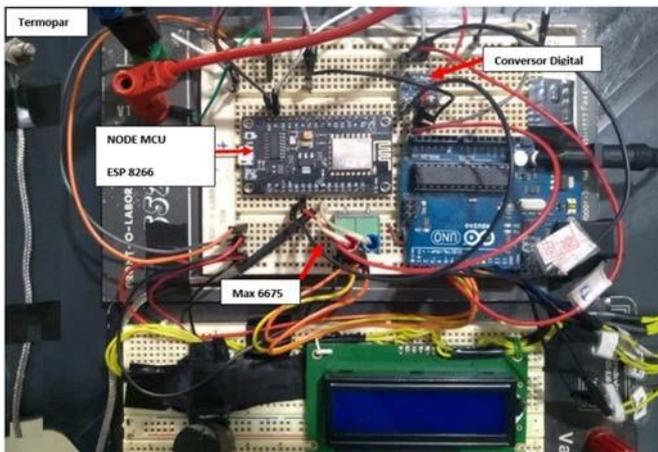


Figura 15 – Diagrama de blocos do protótipo

IV. RESULTADOS E DISCUSSÕES

Foi realizada a programação do Arduino para realizar as leituras das variáveis temperaturas nos módulos fotovoltaicos de acordo com a programação utilizada na simulação. Foi acrescida a função de comunicação com o NODE MCU, através da porta serial estabelecida pela biblioteca SoftwareSerial.h, utilizando os pinos 2 (RX) e 3 (TX) do Arduino Uno e pinos D7 (RX) e D8 (TX) no NODE MCU ESP 8266.

Para desenvolver a programação do Arduino [Apêndice 3], foram utilizadas as bibliotecas: LiquidCrystal.h [Anexo 1], max6675.h [Anexo 2] e SoftwareSerial.h [Anexo 3]. Não foi desenvolvido um protocolo específico para a comunicação entre o Arduino e o NODE MCU, vez que, a mesma ocorre através das portas seriais e o sincronismo realizado pelo reset das duas placas. Desta forma, a comunicação se estabelece e os dados das variáveis são transferidos para o NODE MCU.

A programação do NODE MCU ESP 8266 [Apêndice 4], foi desenvolvida para receber os dados de temperatura dos módulos fotovoltaicos e estabelecer a comunicação com o servidor de aplicação AllThingsTalk Maker, possibilitando o acesso desses dados através da Internet. Nesta programação foi utilizado as bibliotecas SoftwareSerial.h [Anexo 3] e AllThingsTalk_WiFi.h [Anexo 4]. O NODE MCU ESP estabelece a comunicação com o aplicativo AllThingsTalk Maker, através do WiFi, e os dados de temperatura dos módulos Fotovoltaicos são transferidos para a Nuvem. A Figura 16 abaixo, apresenta, no monitor serial da IDE do Arduino, os status de conexão com o WiFi da rede utilizada e os dados transferidos do NODE MCU para a Nuvem.

```
23:15:36.633 -> | TX|K|W|I|F|I|P|R|I| |
23:15:36.834 -> ----- AllThingsTalk WiFi SDK Serial Begin -----
23:15:36.902 -> Debug Level: Normal
23:15:36.902 -> Connection LED: Enabled - GPIO 2
23:15:36.936 -> WiFi Signal Reporting: Enabled
23:15:36.970 -> Connecting to WiFi: Alencar Oi 2.4G.....
23:15:42.459 -> Connected to WiFi!
23:15:43.258 -> Your device can't be created on AllThingsTalk because the HTTP Connection failed.
23:15:47.144 -> Connecting to AllThingsTalk.
23:15:47.660 -> Connected to AllThingsTalk!
23:15:48.274 -> Message Published to AllThingsTalk (JSON)
23:15:49.265 -> TEMPERATURA MÓDULO1 38.79
23:15:49.265 -> Message Published to AllThingsTalk (JSON)
23:15:50.255 -> TEMPERATURA MÓDULO1 38.79
23:15:50.255 -> Message Published to AllThingsTalk (JSON)
23:15:51.247 -> TEMPERATURA MÓDULO1 38.79
23:15:51.247 -> Message Published to AllThingsTalk (JSON)
```

Figura 16 – Status da conexão com a rede Wi-Fi e demonstração da transferência dos dados medidos para a Nuvem

No Hardware desenvolvido é possível visualizar, em um Display LCD, a variável temperatura do módulo Fotovoltaico. Na Figura 17 abaixo, tem-se a representação da variável de temperatura do Módulo 1 medida através desta metodologia.



Figura 17 – Valores medidos de temperatura no Módulo 1, demonstrados no LCD do sistema construído

Uma vez estabelecido a conexão do NODE MCU ESP 8266 com a API de Nuvem AllThingsTalkMaker, os dados transferidos são recebidos e armazenados na mesma. Os dados enviados podem ser acompanhados em tempo real através da página do aplicativo. A Figura 18 abaixo, apresenta o gráfico da temperatura do Módulo 1 no formato de barras.

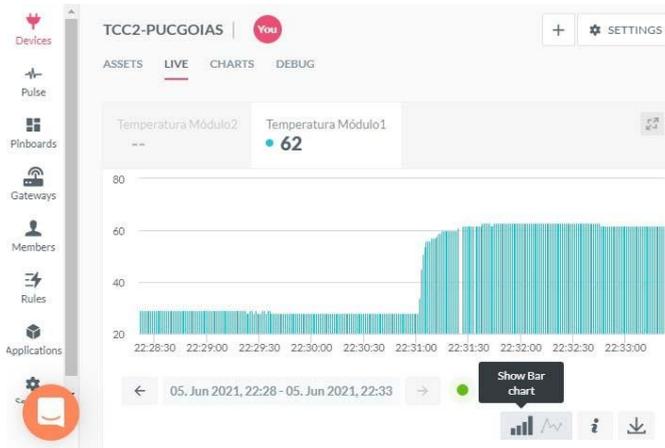


Figura 18 – Temperatura, em tempo real, no aplicativo do AllThingsTalk Maker, com representação em gráfico de barras

O aplicativo disponibiliza ainda, diferentes opções para a organização dos dados coletados. Como, por exemplo, no formato de gráfico de linha e, até mesmo, com intervalos nas medições para melhor visualização dos valores medidos. Conforme demonstrado na Figura 19, abaixo.

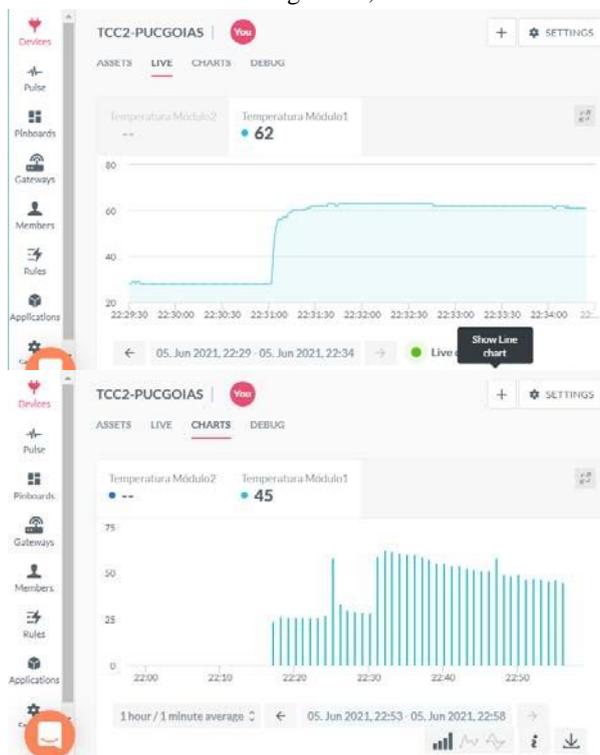


Figura 19 – Temperatura, em tempo real, no aplicativo do AllThingsTalk Maker, com representação, respectivamente, em gráfico de linha e gráfico em barras com intervalo de um minuto entre as medições

No aplicativo é possível estabelecer regras e ações de controle que possam ser devolvidas para o chão de fábrica, estabelecendo uma rede de sensores e atuadores interligadas para aplicações em IoT. E, Conforme ilustrado nesta seção, o estabelecimento de uma conexão com a Nuvem, utilizando tecnologias de IoT, pode ser realizada através do Hardware proposto e da API AllThingsTalkMaker.

Os resultados foram satisfatórios e podem ser utilizados para várias aplicações de IoT. Pode ser observado que a transmissão dos dados é segura e reflete o que realmente ocorre ao nível dos sensores. Foi realizada uma alteração na temperatura do sistema e o sistema apresentou em tempo real a mudança ocorrida. Isto pode ser observado nos gráficos da Figura 18 e 19. A versão gratuita do aplicativo pode monitorar até 10 sensores, que para trabalhos acadêmicos, pode ser uma alternativa interessante.

V. CONCLUSÃO

Este trabalho apresenta um estudo sobre a implementação de um protótipo capaz de agrupar uma série de sensores e atuadores, que realizarão medições e disponibilizarão esses resultados, disponíveis para a análise em tempo real, possibilitando a tomada de decisão de forma consciente e embasada em dados estatísticos sobre a realidade do sistema monitorado. O protótipo utiliza do Módulo NodeMCU ESP8266 12-E que realiza a conexão do protótipo com a Nuvem através do aplicativo AllThingsTalk Maker, utilizando da conexão Wi-Fi. A ideia de se utilizar sensores inteligentes que apresentem dados sobre a situação real de sistemas já é difundida mundialmente em setores distintos, especialmente as Cidades Inteligentes e Construções Inteligentes.

O foco de suas aplicações é se aproveitar da capacidade de automatização dos processos e da facilidade no monitoramento dos mesmos, o que pode ser observado neste estudo através de um cenário simulado. Em detrimento da realidade enfrentada da pandemia do Covid-19, não foi possível implementar o protótipo em sistemas fotovoltaicos reais, porém, o estudo foi capaz de comprovar a facilidade no monitoramento da temperatura de módulos fotovoltaicos, em tempo real e, através da Internet.

A plataforma utilizada, disponibilizada pelo AllThingsTalk, consegue monitorar simultaneamente até 10 sensores, além de permitir também, a implementação de atuadores, que possibilitam a automatização de tarefas mais complexas. O estudo se comprometeu em avaliar a disponibilidade de aplicação de um protótipo que capacitasse o usuário a realizar o monitoramento e a análise em tempo real dos parâmetros de temperatura de um sistema fotovoltaico, em detrimento da influência dessa grandeza física na produção de energia dos sistemas. Os resultados mostram a coleta em tempo real desses valores de temperatura do ambiente onde foi instalado o protótipo, tendo obtido êxito no propósito determinado de construí-lo e comprovar seu correto funcionamento.

VI. REFERÊNCIAS

- [1] ANTHONY, S. **World's first road-powered electric vehicle network switches on in South Korea**. Publicado em 6 de agosto de 2013. Disponível em: < <https://www.extremetech.com/extreme/163171-worlds-first-road-powered-electric-vehicle-network-switches-on-in-south-korea>>. Acesso em: 26 nov. 2020.
- [2] BRANQUINHO, O. C.; REGIANNI, N.; ANDREOLLO, A. G. **Redes de comunicação de dados sem fio – uma análise de desempenho**. Disponível em: <<http://www.homexpert.com.br/areaservada/wp-content/uploads/2011/06/Redes-de-comunica%C3%A7%C3%A3o-sem-fio-Uma-an%C3%A1lise-de-desempenho.pdf>>. Acesso em: 26 nov. 2020.
- [3] **Redes Sem Fio: Tecnologias de Redes Sem Fio**. Seção de tutoriais sobre Redes Ópticas no site da Teleco. Disponível em: <https://www.teleco.com.br/tutoriais/tutorialredespbaid/pagina_3.asp>. Acesso em: 26 nov. 2020.
- [4] PEREIRA, E. B.; ABBADE, A. L. R. **Comparativo rede PON x rede mista**. V SRST - Seminário de Redes e Sistemas de Telecomunicações, set. 2016. Disponível em: <<https://www.google.com/url?sa=t&rcrct=j&q=&esrc=s&source=web&cd=1&ved=2ahUKEwjhodDr9pfdAhUmVvKkHYAfBzoQFjAAegQICRAC&url=https%3A%2F%2Fwww.inatel.br%2Fbiblioteca%2Fpos-seminarios%2Fseminario-de-redes-e-sistemas-detelecomunicacoes%2Fv-srst%2F9517-comparativo-rede-pon-x-redemista%2Ffile&usg=AOvVaw2KiYs5Ihe7zFipt7nB1nrz>>. Acesso em: 31 ago. 2018.
- [5] ALECRIM, E. **O que é Wi-Fi (IEEE 802.11)?**. Publicado em 19 de março de 2008 e atualizado em 24 de junho de 2013. Disponível em: < <https://www.infowester.com/wifi.php>>. Acesso em: 26 nov. 2020.
- [6] **Redes Wi-Fi I: Espectro de Frequência ISM**. Seção de tutoriais sobre Banda Larga no site da Teleco. Disponível em: < https://www.teleco.com.br/tutoriais/tutorialredeswifi1/pagina_5.asp>. Acesso em: 26 nov. 2020.
- [7] SILVA, I. L. C. S. **Do 1G ao 5G: Evolução das redes de telefonia móvel**. Cruz das Almas, 2016. Disponível em: < https://www2.ufrb.edu.br/bcet/components/com_chronoforms5/chronofrms/uploads/tcc/20190327163532_2015.2_-_TCC_Itala_Liz_-_Do_1g_Ao_5g_Evolucao_Das_Redes_de_Telefonia_Movel.pdf>. Acesso em: 26 nov. 2020.
- [8] BOLTON, W.; XIAO, Y.; GUIZANI, M. **IEEE 802.20: Mobile Broadband Wireless Access**. Publicado em Março de 2007. Disponível em: < file:///D:/Downloads/802.20_IEEEFeb07.pdf>. Acesso em: 26 nov. 2020.
- [9] Agência Brasil. **Maior leilão da história trará o 5G ao Brasil**. Publicado em Maio de 2021. Disponível em: <<https://www.institutomillennium.org.br/maior-leilao-da-historia-trara-o-5g-ao-brasil/#:~:text=Muito%20mais%20do%20que%20uma,sociedade%20e%20os%20meios%20produtivos.&text=O%20edita%20do%20leil%C3%A3o%205G,at%C3%A9%20o%20fim%20deste%20ano>>. Acesso em: 29 mai. 2021.
- [10] COUTINHO, D. **5G brasileiro ainda pode levar anos para ser implementado; entenda o que falta**. Publicado em: Janeiro de 2021. Disponível em: <<https://tecnologia.ig.com.br/2021-01-09/5g-brasileiro-ainda-pode-levar-anos-para-ser-implementado-entenda-o-que-falta.html>>. Acesso em: 29 mai. 2021.
- [11] **5G: Tecnologias de Celular**. Publicado em Julho de 2020. Disponível em: <https://www.teleco.com.br/5g_tecnologia.asp>. Acesso em: 29 mai. 2021.
- [12] **About 3GPP**. Postagem do site da 3GPP. Disponível em: < <https://www.3gpp.org/about-3gpp>>. Acesso em: 26 nov. 2020.
- [13] SILVA, Cristoffer Leite; BARRETO, Priscila Solís; CAETANO, Marcos Fagundes; ALCHIERI, Eduardo Adílio; SOARES, Rafael Amaral. **Uma Proposta para Avaliação da Virtualização de Funções de Rede em 5G**. In: WORKSHOP DE GERÊNCIA E OPERAÇÃO DE REDES E SERVIÇOS (WGRS), 25. , 2020, Rio de Janeiro. **Anais [...]**. Porto Alegre: Sociedade Brasileira de Computação, 2020 . p. 263-276. ISSN 2595-2722. DOI: <https://doi.org/10.5753/wgrs.2020.12466>.
- [14] **3GPP Low Power Wide Area Technologies**. GSMA White Paper publicado em 2016. Disponível em: < <https://www.gsma.com/iot/wp-content/uploads/2016/10/3GPP-Low-Power-Wide-Area-Technologies-GSMA-White-Paper.pdf>>. Acesso em: 26 nov. 2020.
- [15] **GSMA Welcomes 3GPP Completion of LPWA Technologies**. Publicação do portal de notícias do site da GSMA, datado em 23 de junho de 2016. Disponível em: < <https://www.gsma.com/newsroom/press-release/gsma-welcomes-3gpp-completion-lpwa-technologies/>>. Acesso em: 26 nov. 2020.
- [16] THOMSEN, A. **O que é Arduino?**. Publicado em: Setembro de 2014. Disponível em: <<https://www.filipeflop.com/blog/o-que-e-arduino/>>. Acesso em: 29 mai. 2021.
- [17] BAUERMEISTER, G. **Biblioteca Arduino: aprenda como criar a sua – Parte 1**. Publicado em: Abril de 2018. Disponível em: <<https://www.filipeflop.com/blog/desenvolvendo-uma-biblioteca-arduino/>>. Acesso em: 29 mai. 2021.
- [18] **Arduino Energy Monitoring Library**. Publicado em: Janeiro de 2014. Disponível em: <<https://github.com/openenergymonitor/EmonLib/>>. Acesso em: 29 mai; 2021.
- [19] THOMSEN, A. **Como programar o módulo ESP8266 NodeMCU**. Publicado em: Fevereiro 2016. Disponível em: <<https://www.filipeflop.com/blog/esp8266-nodemcu-como-programar/>>. Acesso em: 29 mai. 2021.
- [20] **Como funcionam Servidores de Aplicação?**. Publicado em: Dezembro de 2019. Disponível em: <<https://cl9.com.br/como-funcionam-servidores-de-aplicacao/>>. Acesso em: 29 mai. 2021.
- [21] **Get Started with the ESP8266 (NodeMCU)**. Disponível em: <<https://www.allthingstalk.com/faq/allthingstalk-and-esp8266>>. Acesso em: 29 mai. 2021.
- [22] MULTIEDRO. **Para que serve o Google Cloud Platform e quais as vantagens de utilizá-lo?**. Publicado em: Março de 2020. Disponível em: <<https://blog.multiedro.com.br/para-que-serve-o-google-cloud-platform/>>. Acesso em: 29 mai. 2021.
- [23] **O que é Azure?**. Disponível em: <<https://azure.microsoft.com/pt-br/overview/what-is-azure/>>. Acesso em: 29 mai. 2021.
- [24] DURBANO, V. **Computação em nuvem: tudo que você precisa saber sobre**. Publicado em: 2018. Disponível em: <<https://blog.ecoict.com.br/computacao-em-nuvem/>>. Acesso em: 29 mai. 2021.
- [25] **O que é computação em nuvem?**. Disponível em: <<https://azure.microsoft.com/pt-br/overview/what-is-cloud-computing/#cloud-computing-models>>. Acesso em: 29 mai. 2021.

VII. ANEXOS E APÊNDICES

Apêndice 1 – Passo a passo dos processos de comunicação do NodeMCU ESP8266 com o AllThingsTalk Maker

O passo a passo para implementar o AllThingsTalk Maker na placa ESP8266 (sendo possível a utilização de outros módulos, conforme diagrama da Figura 9) foi disponibilizado no portal da AllThingsTalk e, segue discriminado abaixo, com a ilustração através de recortes feitos durante o processo.

1. Para programação do arduino utilizou-se o software Arduino IDE (Integrated Development Environment) 1.18.13 que pode ser baixado de graça no site <https://www.arduino.cc/en/main/software>;
2. Realizado o download do software, é necessário baixar o "core" da placa ESP8266, uma vez que o Arduino IDE precisa dessas preferências para poder programar a placa:
 - a. Para isso, vá na aba Arquivo > Preferências (ou utilize o comando Ctrl + vírgula) e insira o seguinte URL (http://arduino.esp8266.com/stable/package_esp8266com_index.json) aos URLs Adicionais para Gerenciador de Placa (conforme destaque na Figura 1 abaixo):

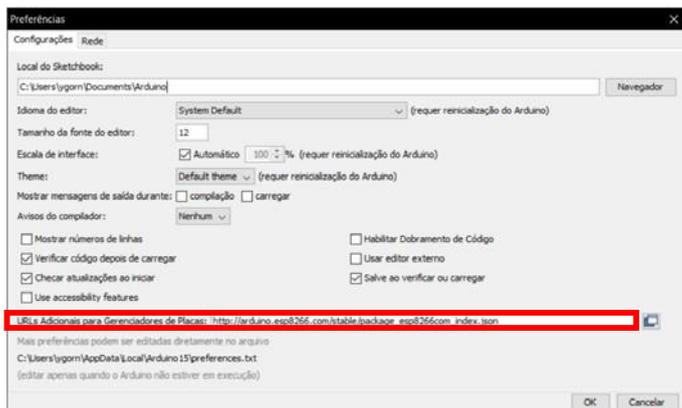


Figura 1 - Recorte da janela de Preferências, disponível através do endereço Arquivos > Preferências ou do comando Ctrl + Vírgula

- b. Confirme em OK para fechar a janela;
- c. Abra o Gerenciado de Placas via Ferramentas > Placa: "Arduino Uno" > Gerenciador de Placas e nessa janela, pesquise e instale o pacote ESP8266, versão 2.74, desenvolvido pela Comunidade ESP8266. Conforme Figura 2 abaixo:



Figura 2 - Recorte da janela do Gerenciador de Placas, disponível através do endereço Ferramentas > Placa: "Arduino Uno" > Gerenciador de Placas

- d. Instalado o pacote, feche a janela do Gerenciador de Placas.
3. Para execução do programa esboço que será utilizado, é necessário a instalação de algumas bibliotecas:
 - a. Vá em Ferramentas > Gerenciador de Biblioteca (ou utilize o comando Ctrl + Shift + I) e nessa janela, pesquise e instale o All Things Talk WiFi SDK versão 2.1.2, desenvolvido pela All Things Talk; Pesquise e instale também, o ArduinoJson versão 6.17.13, desenvolvido por Benoit Blanchon. Conforme Figuras 3 e 4 abaixo:

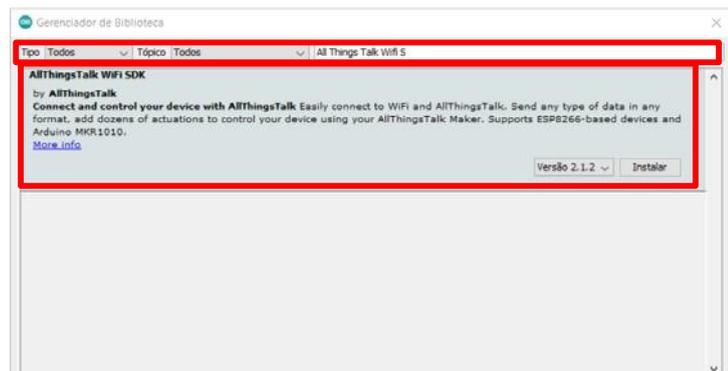


Figura 3 - Recorte da janela do Gerenciador de Placas, disponível através do endereço Ferramentas > Placa: "Arduino Uno" > Gerenciador de Placas

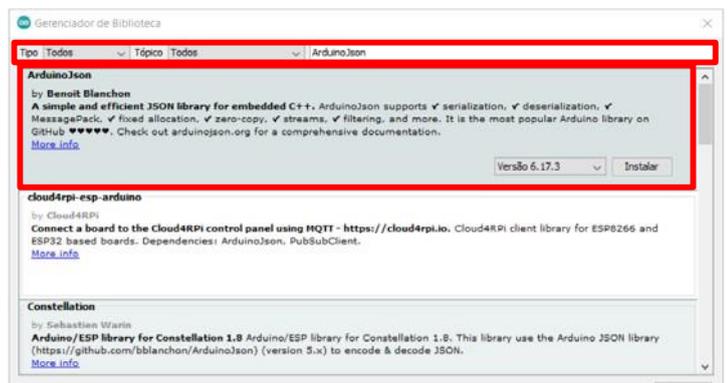


Figura 4 - Recorte da janela do Gerenciador de Bibliotecas, disponível através do endereço Ferramentas > Gerenciador de Bibliotecas

- b. Instalado as bibliotecas, feche a janela do Gerenciador de Bibliotecas.
4. Feito isso, é necessário criar um dispositivo AllThingsTalk Maker. Para isso:
 - a. Cadastre-se no AllThingsTalkMaker através do link: <https://maker.allthingstalk.com/signin>. O cadastro/login pode ser feito utilizando diretamente sua conta do Google;
 - b. Estando logado na plataforma, selecione seu terreno onde será criado um novo dispositivo (toda conta recém criada possui um terreno chamado "Playground" automaticamente).

Clique para conectar um novo dispositivo e selecione o bloco Arduino;

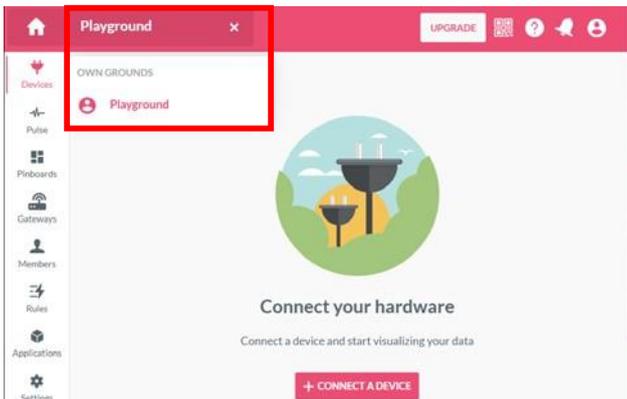


Figura 5 - Recorte da página do AllThingsTalk Maker destacando os locais onde se encontra os “terrenos” e adiciona novos dispositivos

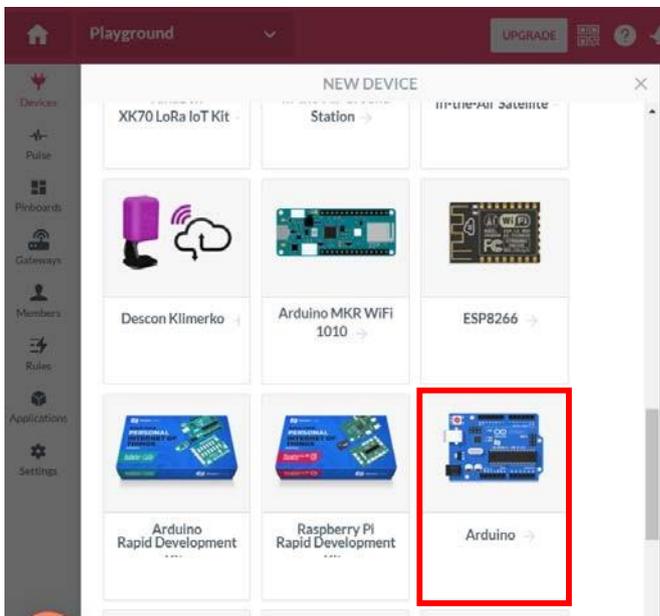


Figura 6 - Recorte da página do AllThingsTalk Maker destacando os dispositivos disponíveis para serem adicionados, em destaque o Arduino.

- c. Com isso, foi inserido seu primeiro dispositivo no All Things Talk Maker, o Arduino que irá receber o programa para conexão com o All Things Talk através da internet;
- d. Para dar continuidade a programação do dispositivo, vamos para o software Arduino IDE e iremos adicionar o programa esboçado pela própria All Things Talk para conexão da placa ESP8266, através do Arduino, com o AllThingsTalk, utilizando o Wi-Fi disponível do NodeMCU – ESP8266.

5. Configura-se, portanto, o seguinte no Arduino IDE (consultar o Anexo 1):
6. A respeito do programa acima, importante apontar que:

- a. Os comentários a respeito dos comandos utilizados no programa se encontram após cada “//”;
- b. É necessário que o usuário, no momento da aplicação do programa, altere o seguinte:
 - i. Altere Your_WiFi para o nome da sua rede WiFi utilizada;
 - ii. Altere Your_WiFi_Password para a senha de sua rede WiFi;
 - iii. Altere Your_Device_ID para o ID do seu dispositivo que pode ser obtido nas configurações do dispositivo em questão no AllThingsTalk, através do endereço Configurações > Autenticação e copie o valor de ID do dispositivo (conforme destaque nas figura abaixo);

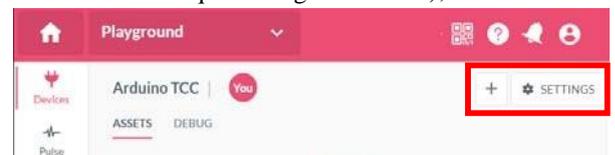


Figura 7 - Recorte da página do AllThingsTalk Maker destacando o local para acessar as configurações dos dispositivos

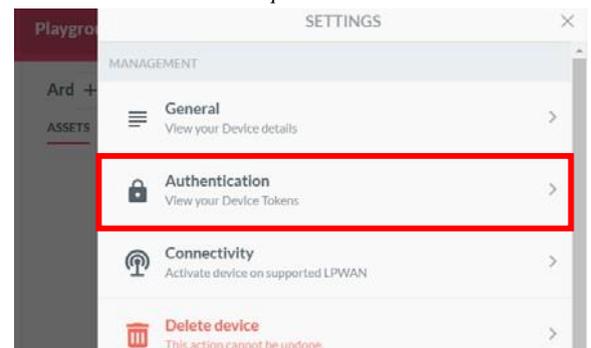


Figura 8 - Recorte da página do AllThingsTalk Maker com o quadro de configurações dos dispositivos, destacando a aba de Autenticação.

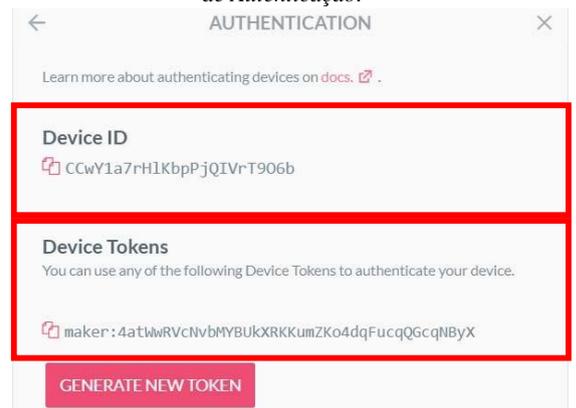


Figura 9 - Recorte da página do AllThingsTalk Maker com o quadro de Autenticação, em destaque o ID e o Token do dispositivo.

- iv. Altere Your_Device-Token para o token do seu dispositivo no AllThingsTalk que pode ser obtido nas configurações do dispositivo em questão no AllThingsTalk,

através do endereço Configurações > Autenticação e copie o token do dispositivo (conforme destaque na figura abaixo)

- v. Feito isso, salva-se as alterações realizadas no programa do Arduino IDE.

7. Agora, para testar o funcionamento do programa acima, será carregado o mesmo na placa ESP8266:

- a. Conecte um cabo USB na entrada existente na ESP8266 (NodeMCU) à uma entrada USB do seu computador;
- b. Vá em Ferramentas > Placa: “Arduino Uno” > ESP8266 Boards (2.7.4) > NodeMCU 1.0 (ESP-12E Module) – ou outro, que será determinado pelo modelo da placa ESP8266 que está sendo utilizada). Conforme Figura 10 abaixo:

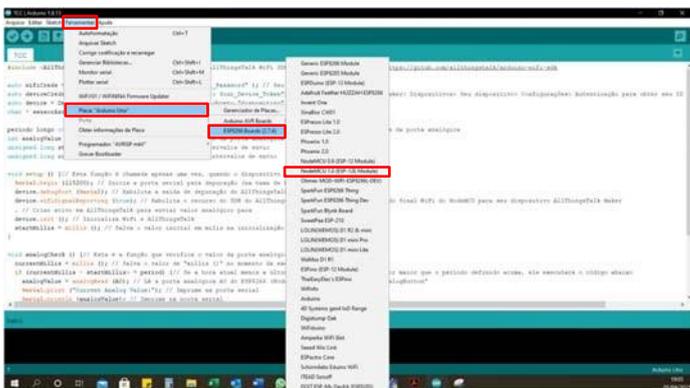


Figura 10 - Recorte do Arduino IDE demonstrando o caminho para definir o modelo da placa a ser feito o carregamento do programa do dispositivo.

- c. Ainda no Arduino IDE, vá em Ferramentas > Portas e, selecione a porta em que o dispositivo está (deve ser a única mostrando no software);

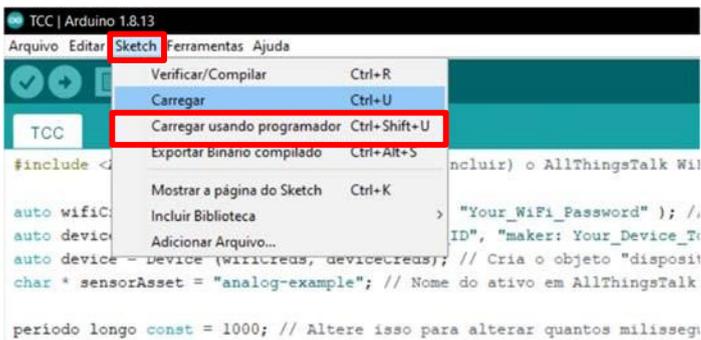


Figura 11 - Recorte do Arduino IDE demonstrando o caminho para realizar o carregamento do programa no dispositivo.

- d. Após isso, clique em Sketch > Carregar para fazer o upload do programa, ou utilize o comando Ctrl + U. Conforme Figura 11 acima.

8. Indicado, na parte inferior do Arduino IDE, que a compilação do esboço começou a carregar na placa ESP8266 (NodeMCU) é necessário esperar a tarefa

finalizar, o que ocorrerá quando aparecer a mensagem “Carregado”, indicando que o código foi enviado para o NodeMCU e está tudo pronto para executá-lo. É demonstrado também, outras informações sobre o módulo e sobre a conexão estabelecida, conforme Figura 12 abaixo.

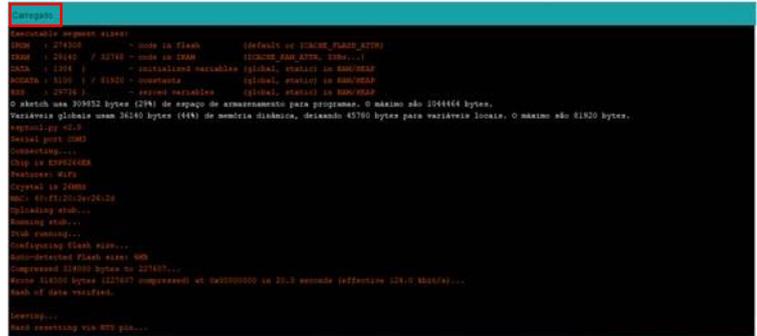


Figura 12 - Recorte do Arduino IDE demonstrando o resultado do carregamento do programa no módulo ESP8266, através da conexão USB ao computador.

9. Defina a taxa de transmissão do seu monitor serial para 115200, através do endereço Ferramentas > Monitor serial, ou utilizando o comando Ctrl + Shift + M. Conforme Figura 13 abaixo;

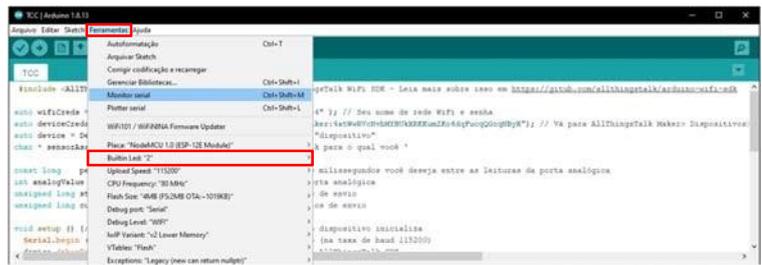


Figura 13 - Recorte do Arduino IDE demonstrando o endereço para o Monitor Serial onde será configurado a taxa de transmissão.

10. A indicação de que a placa ESP8266 está conectada a internet ocorrerá através da mensagem “Conectado ao AllThingsTalk” no monitor serial ou, através do LED azul presente na placa que deverá piscar continuamente, permanecendo dessa forma conforme a Figura 14 abaixo;



Figura 14 – Fotografia retirada da placa funcionando, com conexão USB ligada a porta do notebook e, com o LED Azul piscando indicando seu funcionamento.

11. Com isso, seu dispositivo está efetivamente conectado a internet através do AllThingsTalk, utilizando o Wi-Fi para estabelecer essa conexão;

Apêndice 2 – Software de Programação do Arduino UNO para simulação no ISIS PROTEUS

```
#include <LiquidCrystal.h> //Biblioteca para tratamento do Display LCD

#define MAX6675_CS 5
#define MAX6675_SO 6
#define MAX6675_SCK 4
LiquidCrystal lcd(7, 8, 9, 10, 11, 12);

void setup(){
  Serial.begin(9600); //INICIALIZA A SERIAL PARA O MONITOR SERIAL
  lcd.begin(16, 2);
  delay(2000);
}

void loop(){
  float T = readThermocouple(); // chama a função de Leitura da Temperatura
  Serial.print("TEMPERATURA DO MODULO FV1 = ");
  Serial.println(T);

  //Visualização no LCD

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("TEMPERATURA FV1");
  lcd.setCursor(0, 1);
  lcd.print(T);
  delay(1000);
}

//Função de Leitura do sensor de temperatura MAX6675-Protocolo SPI
double readThermocouple() {

  uint16_t v;
  pinMode(MAX6675_CS, OUTPUT);
  pinMode(MAX6675_SO, INPUT);
  pinMode(MAX6675_SCK, OUTPUT);

  digitalWrite(MAX6675_CS, LOW);
  delay(1);
```

```

v = shiftIn(MAX6675_SO, MAX6675_SCK, MSBFIRST);
v <<= 8;
v |= shiftIn(MAX6675_SO, MAX6675_SCK, MSBFIRST);
digitalWrite(MAX6675_CS, HIGH);
if (v & 0x4)
{
  //O bit 2 indica se o termopar está desconectado
  return NAN;
}

// Os três bits inferiores (0,1,2) são bits de status descartados
v >>= 3;

//Os bits restantes são o número de contagens de 0,25 graus (C)
return v*0.25;
}

```

Apêndice 3 – Software de Programação do Arduino UNO para comunicação com o NODE MCU 8266 ESP-12E

```

#include <SPI.h>
#include "max6675.h"
#include <LiquidCrystal.h>
#include <SoftwareSerial.h>

int thermoDO = 6; // SO = Digital 6
int thermoCS = 5; // CD = Digital 5
int thermoCLK = 4; // SCK = Digital 4

MAX6675 thermocouple(thermoCLK, thermoCS, thermoDO);
SoftwareSerial EspSerial(2,3); // RX, TX Para comunicação com o NODE MCU ESP 8266-12E
LiquidCrystal lcd(7, 8, 9, 10, 11, 12);

void setup(){
  Serial.begin(9600); //INICIALIZA A SERIAL PARA O MONITOR SERIAL
  EspSerial.begin(115200); // INICIALIZA A SERIAL DE COMUNICAÇÃO COM NODE MCU
  lcd.begin(16, 2);
  lcd.clear();
  delay(2000);
}

```

```

void loop(){
  float T = thermocouple.readCelsius(); // Leitura do sensor de Temperatura
  //lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("TEMPERATURA");
  lcd.setCursor(0, 1);
  lcd.print(T);
  delay(1000);
  EspSerial.println(T);
}

```

Apêndice 4 – Software de Programação para o NODE MCU ESP8266-12E

```

#include <AllThingsTalk_WiFi.h>
#include "SoftwareSerial.h"

#define D7 13 //RX
#define D8 15 //TX

const long periodo = 1000;
unsigned long startMillis;
unsigned long currentMillis;
int TempEnvia1 = 0;
int TempEnvia2 = 0;

auto wifiCreds = WifiCredentials ("Alencar Oi 2.4G", "Carloshenrique" );
auto          deviceCreds          =          DeviceConfig          ("t99JeEbgBBJ3e25SkDGuhbbz",
"maker:4vCAnCIMrk4PhfNaKUpckziYSK07xxauB9Bu677");
auto device = Device(wifiCreds, deviceCreds);
char* Temp1Asset = "Modulo1";
char* Temp2Asset = "Modulo2";

SoftwareSerial EspSerial(D7, D8);

void setup() {
  pinMode(D7,INPUT);
  pinMode(D8,OUTPUT);
  EspSerial.begin(115200);
  Serial.begin(9600);
  device.debugPort(Serial);
}

```

```

device.wifiSignalReporting(true);
device.createAsset(Temp1Asset, "Temperatura Módulo1", "sensor", "integer");
device.createAsset(Temp2Asset, "Temperatura Módulo2", "sensor", "integer");
device.init();
startMillis = millis();
}

```

```

void loop() {
device.loop();
EnviaMod1();
EnviaMod2();
}

```

```

void EnviaMod1() {
currentMillis = millis();
if (currentMillis - startMillis >= periodo) {
float valor_analog_lm35 = float(analogRead(A0));
float tensao = ((valor_analog_lm35) * 3.1)/1023;
float Temperatura1 = tensao / 0.010;
TempEnvia1 = int(Temperatura1);
Serial.print("TEMPERATURA MÓDULO1 ");
Serial.println(Temperatura1);
device.send(Temp1Asset, TempEnvia1);
startMillis = currentMillis;
}
}

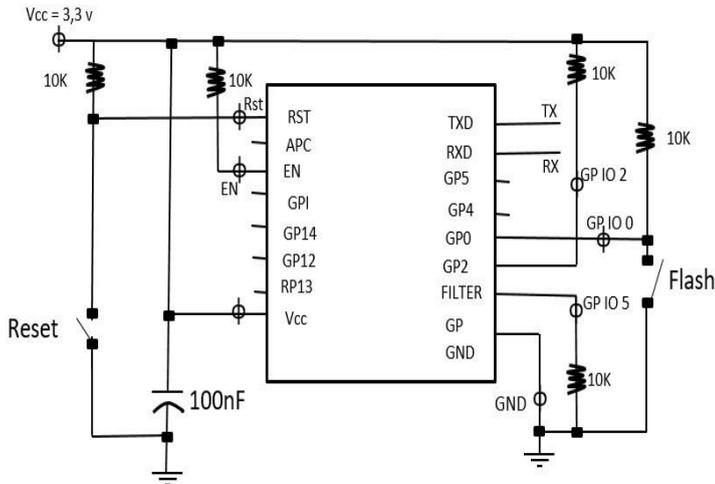
```

```

void EnviaMod2() {
if(EspSerial.available()) {
float(Temperatura2) = EspSerial.parseFloat();
Serial.print(" TEMPERATURA DO MÓDULO 02: ");
Serial.println(Temperatura2);
TempEnvia2 = int(Temperatura2);
device.send(Temp2Asset, TempEnvia2);
}
}

```

Apêndice 5 – Diagrama de conexão do Módulo NODE MCU ESP 8266-12E



OBS: No [arduino](#) fechas (RST-GND)

Anexo 1 – Biblioteca LiquidCrystal

- **LiquidCrystal lcd(rs, en, d4, d5, d6, d7):** Cria uma variável do tipo LiquidCrystal denominada lcd . A exibição é controlada usando 4 linhas de dados (d4, d5, d6 e d8) e os pinos rs e en do display;
- **lcd.begin(col, lin):** Inicializa a interface para o display LCD e especifica as dimensões (coluna e linha) do LCD;
- **lcd.clear():** Apaga todas as informações no LCD;
- **lcd.setCursor(col, lin):** Posiciona o cursor na coluna col e linha lin;
- **lcd.print("text"):** Imprime o texto especificado na string text.

Anexo 2 – Biblioteca Max6675.h

- **MAX6675 thermocouple(CLK, CS, DO):** Cria uma variável denominada thermocouple associada aos pinos de comunicação digital com o protocolo SPI (CLK, CS e DO) ligados aos respectivos pinos do CI;
- **thermocouple.readCelsius():** Fornece a temperatura medida em Graus Celsius.

Anexo 3 – Biblioteca SoftwareSerial.h

- **SoftwareSerial EspSerial(2,3):** Cria uma instância de objeto SoftwareSerial, com nome EspSerial nos pinos digitais 2 e 3 (Rx,Tx).

Anexo 4 – Biblioteca AllThingsTalk_WiFi.h

- <https://github.com/allthingstalk/arduino-wifi-sdk>

RESOLUÇÃO n°038/2020 – CEPE

ANEXO 3

Termo de autorização de publicação de produção acadêmica

Os estudantes, João Vítor Prado Rodrigues, do Curso de Engenharia Elétrica matrícula 2016.1.0038.0245-8, telefone: (62) 98285-3796 e-mail joavitorprado18@gmail.com e Ygor Nadler da Silva, do Curso de Engenharia Elétrica matrícula 2016.1.0038.0182-6, telefone: (62) 98214-0433 e-mail ygor.nadler.2012@gmail.com, nas qualidades de titulares dos direitos autorais, em consonância com a Lei n° 9.610/98 (Lei dos Direitos do autor), autorizam a Pontifícia Universidade Católica de Goiás (PUC Goiás) a disponibilizar o Trabalho de Conclusão de Curso intitulado **PROTÓTIPO PARA MONITORAMENTO DOS PARÂMETROS DE TEMPERATURA EM MÓDULOS FOTOVOLTAICOS POR INTERNET DAS COISAS – IoT**, gratuitamente, sem ressarcimento dos direitos autorais, por 5 (cinco) anos, conforme permissões do documento, em meio eletrônico, na rede mundial de computadores, no formato especificado (Texto (PDF); Imagem (GIF ou JPEG); Som (WAVE, MPEG, AIFF, SND); Vídeo (MPEG, MWV, AVI, QT); outros, específicos da área; para fins de leitura e/ou impressão pela internet, a título de divulgação da produção científica gerada nos cursos de graduação da PUC Goiás.

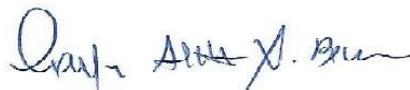
Goiânia, 09 de junho de 2021.



João Vítor Prado Rodrigues



Ygor Nadler da Silva



Professor-orientador: Carlos Alberto Vasconcelos Bezerra