

Liceu de Artes e Ofícios de São Paulo

Gustavo Higuti de Souza

Ian Aguila Sanchez

Igor de Lucca Carneiro Antonio

**CloudIO: Sistema de medição de temperatura e umidade com
armazenamento em nuvem**

São Paulo

2020

Gustavo Higuti de Souza
Ian Aguila Sanchez
Igor de Lucca Carneiro Antonio

**CloudIO: Sistema de medição de temperatura e umidade com
armazenamento em nuvem**

Trabalho de conclusão de curso
apresentado ao Liceu de Artes e Ofícios de
São Paulo requisito parcial para a obtenção
do Título de Técnico em Eletrônica.

Orientador: Milton Barreiro Júnior

São Paulo

2020

Gustavo Higuti de Souza
Ian Aguila Sanchez
Igor de Lucca Carneiro Antonio

**CloudIO: Sistema de medição de temperatura e umidade com
armazenamento em nuvem**

Monografia apresentada à escola Liceu de
Artes e Ofícios de São Paulo como requisito
parcial para a obtenção do Título de
Técnico em Eletrônica.

X

Milton Barreiro Júnior
Orientador

São Paulo
2020

Gustavo Higuti de Souza
Ian Aguila Sanchez
Igor de Lucca Carneiro Antonio

**CloudIO: Sistema de medição de temperatura e umidade com
armazenamento em nuvem**

Monografia julgada e aprovada:

Professor Orientador: Milton Barreiro Junior _____

Professor Coorientador: MSc. Hugo da Silva B. Gonçalves _____

Coordenador de curso: MSc. Sérgio Minas Melconian _____

São Paulo

2020

DEDICATÓRIA

Às nossas famílias, professores e amigos,
que nos acompanharam em todo o trajeto.
E à amizade que uniu este tão querido grupo.

AGRADECIMENTOS

Agradecemos ao colégio Liceu de Artes e Ofícios de São Paulo, instituição que nos acolheu nos últimos anos, por dar-nos esta oportunidade, aos nossos amigos e colegas, que se mantiveram unidos mesmo sob as mais difíceis circunstâncias e a todos os professores que nos auxiliaram em nossa caminhada, que se esforçaram ao máximo em se adaptar ao ano atípico no qual esse projeto foi realizado, se atualizando e aprendendo dia após dia. Por fim, temos um agradecimento especial aos mentores Milton Barreiro, Hugo Bernardes e Aylton de Souza que estiveram sempre nos auxiliando e apoiando durante o projeto.

EPÍGRAFE

*“No one has ever changed the world by doing
what the world has told them to do.”*

*(“Ninguém nunca mudou o mundo fazendo
o que o mundo os falou para fazer.”)*

– Eddy Zhong

RESUMO

O projeto realizado é um sistema de monitoramento de informações (como umidade e temperatura de um local) com dados armazenados em nuvem.

Em sua elaboração, foi utilizado um Arduino UNO R3 e um sensor DHT22, para a obtenção das informações de temperatura e umidade de um local, e um sistema montado com um VI (Virtual Instrument) programado em LabVIEW, que é executado em um computador, com integração à alguma nuvem, onde são armazenados os dados em uma planilha na nuvem, cuja visualização dos mesmos pode ser feita de qualquer lugar do mundo, usando qualquer dispositivo com acesso à internet. Para facilitar a visualização, foi criado um modelo de dashboard para melhorar a exibição dos dados, tornando a experiência mais agradável e intuitiva para o usuário.

Palavras-chaves: Monitoramento, sensoriamento, IoT, Arduino, LabVIEW, Microsoft, PowerBI, Excel, Internet e DHT22

ABSTRACT

The realized project is an information-monitoring system (information such as humidity and temperature of a spot) with the data stored in the cloud.

In its elaboration, it was utilized an Arduino UNO R3 and a sensor DHT22, for the data acquisition of the temperature and humidity of a spot, and a system made with a VI (Virtual Instrument) programmed in LabVIEW, that is executed in a computer, integrated to a cloud service, where the data is stored at a spreadsheet, in which the data visualization can be made from anywhere in the world, using any device connected to the internet. To facilitate the preview, it was created a dashboard model to enhance the data exhibition, making the experience more pleasant and intuitive to the user.

Key words: Monitoring, sensing, IoT, Arduino, LabVIEW, Microsoft, PowerBI, Excel, Internet e DHT22

ABREVIACES E SIGLAS

VI Virtual Instrument (instrumento virtual)

IoT Internet of things (do ingls, internet das coisas)

LISTA DE FIGURAS

Figura 1 – Imagem de um sensor DHT22	Pág 17
Figura 2 – Sensores internos de um DHT22	Pág 17
Figura 3 – Imagem de um módulo GSM	Pág 18
Figura 4 – Interface do programa no Lab View	Pág 25
Figura 5 – Listamento dos estados do programa	Pág 25
Figura 6 – Programa em estado “Idle”	Pág 26
Figura 7 – Programa em estado “Open serial”	Pág 26
Figura 8 – Programa em estado “Request”	Pág 26
Figura 9 – Programa em estado “Receive”	Pág 27
Figura 10 – Descrição do bloco “Data rescue”	Pág 27
Figura 11 – Programa em estado “Recording”	Pág 27
Figura 12 – Descrição do bloco “Info Formatter”	Pág 28
Figura 13 – Programa em estado “Close serial”	Pág 28
Figura 14 – Arquivo editado pelo programa	Pág 29
Figura 15 – Representação gráfica na interface do Power BI	Pág 29

LISTA DE TABELAS

Tabela 1 – Orçamento do projeto

Pág 20

SUMÁRIO

1 INTRODUÇÃO	14
1.1 Estado de arte	14
1.2 Objetivo	15
1.3 Motivação	15
1.4 Impeditivos	16
2 METODOLOGIA	16
2.1 Sensores	16
2.2 Plataforma de prototipagem	17
2.3 Programação	18
2.4 Comunicação	19
2.5 Banco de dados	19
2.6 Orçamento do projeto	20
3 CONCLUSÃO DO PROJETO	20
3.1 Conclusões	20
3.2 Conclusão de hardware.....	21
3.3 Conclusões de software.....	21
3.4 Propostas para projetos futuros	21
CONSIDERAÇÕES FINAIS	23
REFERÊNCIAS	24
APÊNDICES (códigos usados)	25

Capítulo 1

INTRODUÇÃO

Desde a década de 1990, o conceito da conexão à internet vem se tornando cada vez mais popular, devido principalmente à comodidade e facilidade que proporciona ao usuário, permitindo-o adquirir informações advindas de qualquer lugar do planeta em poucos segundos.

Desse modo, da mesma maneira que a internet, o IoT também está se tornando uma parte integrante da vida cotidiana, apresentando incontáveis aplicações nas mais variadas esferas da sociedade, indo do ajuste de temperatura de um ar condicionado até o monitoramento de extensas áreas florestais, o que faz urgir pela necessidade de uma rede de sensoriamento cada vez maior, visando lidar com o aumento da demanda da internet das coisas.

No entanto, a aplicabilidade do IoT é severamente prejudicada em ambientes que técnicos no assunto não se façam presentes, como residências ou comércios de baixa escala, devido ao baixo conhecimento da população geral em relação ao assunto, que necessita de certo expertise para que seja executado de maneira eficiente.

1.1 Estado de arte

Atualmente, no ano de 2020, o conceito de um sistema de sensoriamento com o armazenamento de dados na nuvem ainda é pouco explorado no ambiente industrial, com uma grande parte das empresas desses ramos optando por guardar suas informações em bancos de dados próprios, transportando-a através de sistemas como a ethernet.

No entanto, nos ambientes residenciais e em comércios de pequena escala, a utilização e aplicação desses tipos de sistema apresenta um crescimento consistente, o que resultou, com o passar dos anos, em diversos pesquisas e projetos semelhantes a esse envolvendo a aplicação do IoT.

Esse projeto trata, principalmente, de uma melhora desse tipo de aplicação, principalmente em relação a seu preço e à sua interação interface usuário.

1.2 Objetivo

Levando esses fatores em consideração, decidiu-se que este projeto de TCC tem como principal objetivo monitorar as condições de ambientes residenciais e comerciais, por um menor custo e com a função de armazenar os dados em nuvem, para facilitar o monitoramento por meio da praticidade da internet, além de conter uma interface de fácil acesso, diminuindo a curva de aprendizagem necessária para a leitura da informação.

1.3 Motivação

Por meio de pesquisas, o grupo constatou sistemas de monitoramento à distância com preços mais acessíveis, além de que a maioria desses não realizam suas funções utilizando dados em nuvem, gerando maior dificuldade de aplicação. Esses fatores dificultam sua utilização tanto em ambientes domésticos, que em geral não apresenta técnicos que possam desenvolver essas aplicações, quanto para empresas em processos relativamente simples, sendo muito caros de serem instalados.

1.4 Impeditivos

O projeto foi executado no ano de 2020, em plena pandemia de COVID-19. Portanto, devido ao isolamento social decretado, suspensão de aulas e fechamento de lojas, estratégias de vários trabalhos realizados no ano em questão tiveram de ser repensadas.

Entretanto, o início da idealização deste projeto se deu quando a situação toda já estava instalada e, dessa forma, já foi pensado de modo a ser exequível na situação vigente.

CAPÍTULO 2

METODOLOGIA

Este capítulo será dedicado à explicação da escolha e aplicação de equipamentos, softwares e protocolos que foram utilizados no desenvolvimento do projeto.

2.1 Sensores

Por tratar-se de um projeto de medição e monitoramento, uma das partes mais fundamentais do sistema é o sensoriamento. Desse modo, buscaram-se alternativas para permitir a medição de umidade e temperatura no ambiente. Após as pesquisas elegeu-se o sensor DHT22 como principal sensor do sistema, visto que é capaz de medir umidade e temperatura simultaneamente.

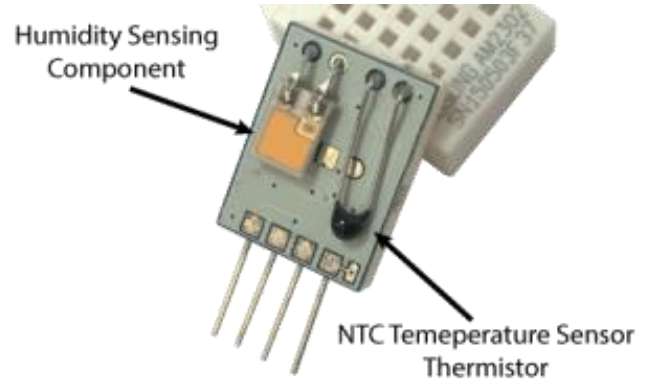
Tendo em sua arquitetura um sensor de umidade capacitivo e um termo resistor, tal componente se enquadra muito bem no projeto, apresentando compatibilidade com o sistema Arduino e também por estar dentro da faixa de preço desejada.

Figura 1: Imagem de um sensor DHT22



Fonte: howtomechatronics.com, acesso em: 07 de ago. 2020, às 16:04.

Figura 2: Sensores internos de um DHT22



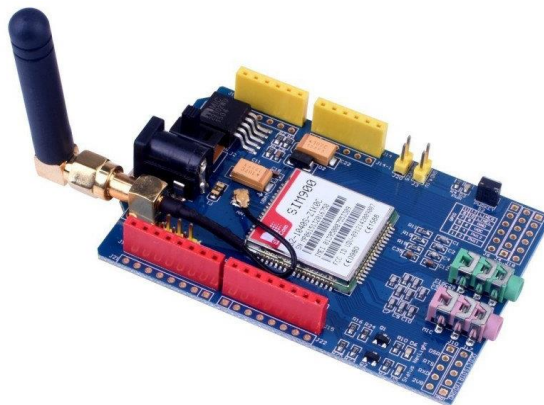
Fonte: www.pcboard.ca, acesso em: 07 de ago. 2020, às 16:23.

2.2 Plataforma de prototipagem

Para que houvesse o desenvolvimento do projeto, foram selecionadas algumas plataformas, sendo essas o Arduino e o LabView.

A plataforma Arduino, teve como princípios tanto a praticidade da programação quanto o baixo custo, elementos esses que são chave para o projeto. No entanto, devido à natureza da plataforma, seriam necessários módulos extras, como o Shield GSM ou o Shield Wi-fi, além de que também haveria necessidade da utilização de serviços externos de nuvem, como o Microsoft Azure ou o Amazon AWS, sendo sua integração no projeto muito árdua.

Figura 3: Imagem de um módulo GSM



Fonte: howtomechatronics.com, acesso em: 07 de ago. 2020, às 15:35.

Portanto, para que fosse resolvido esse problema foi implementado o LabView, que além de apresentar uma interface mais intuitiva, sendo organizado em um esquema de blocos, facilitando a visualização do projeto, permite um controle melhor com relação aos erros e aos dados que estão sendo transportado em cada fase do processo.

2.3 Programação

As principais linguagens de programação utilizada foram C++ (Arduino) e LabVIEW.

O código de programação utilizado no Arduino, demonstrado no Apêndice 2, consiste em um programa que detecta a presença de um sinal enviado pelo LabVIEW e, ao reconhecer a presença deste sinal, lê as medições de temperatura e umidade realizadas pelo sensor. Assim, uma string é mandada seguindo a seguinte estrutura:

“Sinal da temperatura + dezena da temperatura + unidade da temperatura + decimal da temperatura +; + centena da umidade + dezena da umidade + unidade da umidade + decimal da umidade”

Nesse formato o programa consegue detectar níveis de umidade (%) e temperatura (°C) com precisão de 1 até casa decimal.

Após isso, o código de programação em LabVIEW (Apêndice 1) formata novamente os dados requisitados do Arduino, que depois são salvos em um arquivo de texto (.txt), que será lido e traduzido na forma de um gráfico, que poderá ser visualizado através do Power BI.

2.4 Comunicação

Para que houvesse a comunicação entre o Arduino e o computador utilizado havia uma série de possibilidades que poderiam ser implementadas, no entanto, para o desenvolvimento do projeto foi utilizado um cabo USB que, por padrão da plataforma, a comunicação por esse meio é feita utilizando o protocolo UART.

2.5 Banco de dados

Após o processamento dos dados realizado pelo LabVIEW, os dados são salvos na nuvem, na forma de um arquivo de texto (.txt) (figura 13), esse banco de dados cresce conforme o sistema realiza as medições e é armazenado em uma estrutura que permite executar o código em um Power BI, facilitando a visualização dos dados por meio de uma interface de fácil utilização. (figura 14).

O sistema registra no arquivo de texto tanto os dados quanto o momento de sua medição, permitindo localizar o horário de qualquer anomalia detectada, sendo essas de fácil identificação no gráfico. Além disso, o armazenamento dos dados em rede permite o monitoramento dos dados de a partir de qualquer local com acesso à internet, facilitando ainda mais o acompanhamento dos resultados.

2.6 Orçamento do projeto

Para que seja o projeto, foi necessária a compra de aparelhos, sendo suas especificações e preços demonstrados na tabela a seguir.

Tabela 1: Orçamento do projeto

Materiais	Custo em dólares (\$)
Sensor DHT22	9,99
Arduino Uno R3	23,00
Cabo USB	3,95
Notebook HP6UY82UA#ABA	474,95
Total	511,89

Fonte: Autoria própria, 2020

CAPÍTULO 3

CONCLUSÃO DO PROJETO

3.1 Conclusões

Após a finalização do projeto, o grupo conseguiu chegar a algumas conclusões a respeito de cada etapa. Assim, munidos dessas conclusões, esperamos que futuramente, independente da continuação do sistema pelos próprios integrantes do grupo ou por outros indivíduos, que as ideias que deram origem ao projeto possam

ser desenvolvidas e que, futuramente, integrem-se em um produto acessível às massas.

3.2 Conclusão do hardware

Se tratando do hardware, com base na definição dos objetivos do projeto, uma de nossas metas foi fazer com que o conjunto fosse o mais econômico e, simultaneamente, eficiente possível, no entanto, apesar do resultado final satisfatório, sua utilização em ambientes excepcionais àqueles que não apresentam condições extremas, como residenciais e comerciais em geral. Dessa forma, para alcançar um ramo industrial, seriam necessárias adaptações principalmente de hardware

3.3 Conclusão do software

Para que o Power BI pudesse ser implementado no projeto, foi utilizada uma licença estudantil. Assim sendo, apesar de apresentar diversas das funções principais do programa, devido a limitações da mesma, não foi possível automatizar a obtenção de dados, sendo necessária uma atualização manual da interface por parte do usuário.

Com isso em mente, para que houvesse uma maior otimização do projeto, seria necessária a utilização de uma licença profissional do Power BI.

3.4 Propostas para projetos futuros

Este projeto, apesar de ter cumprido com seus objetivos iniciais, abre portas para vastas possibilidades. Algumas delas são:

- Implementar o uso do protocolo MQTT e um sistema de nuvem externo como Amazon AWS, Microsoft Azure ou IBM Watson, de modo a possibilitar a integração a sistemas industriais pré-existentes.
- O uso de outras plataformas como o Excel para a visualização dos dados, ao invés do Power BI.
- Pensando ainda no âmbito industrial, seria válida uma adaptação de hardware para permitir o trabalho em ambientes ruidosos eletromagneticamente.
- O grupo também julga válida uma readaptação de plataforma, utilizando um Raspberry Pi (indica-se a versão Zero W), que poderá substituir tanto o arduino quanto o computador, medindo os dados diretamente do sensor, graças ao seu micro controlador integrado, e criando arquivos e mandando para a nuvem via internet utilizando códigos em Python. Isso tornaria o circuito extremamente mais barato, entretanto é uma mudança radical que levaria bastante tempo para ser desenvolvida, mas poderia, sim, servir como versão final plenamente comercializável do projeto.
- Outra maneira de fazer isso seria utilizar um arduino com Shield Wi-Fi ou GSM para enviar os dados para serviço de nuvem externo, o que também tomaria tempo mas faria o projeto mais comercializável.
- Por fim, a fim de tornar o projeto totalmente escalável e comercializável é válido migrar para uma plataforma micro controlada de arquitetura ARM, fazendo todas as integrações, e criando uma interface que permite usar outros tipos de sensores, além dos de temperatura e umidade, fazendo do projeto algo modular e extremamente mais vendável.

Vale ressaltar que essas propostas são para projetos futuros, e que fogem parcial ou até completamente da premissa inicial deste trabalho, mas são extremamente válidas para uma realização futura.

CONSIDERAÇÕES FINAIS

Portanto, apesar de incluir-se no vasto mercado de sensores, o CloudIO alcança um nicho pouco explorado dessa área, visando a inclusão daqueles que não teriam condição e/ou expertise para se utilizar dessa tecnologia, o que, ultimamente, vem se tornando uma fatia cada vez maior do mercado.

Por fim, apesar de alcançar os metas e objetivos finais do projeto, ainda há muito espaço para melhorias tanto de software quanto de hardware para a criação de um sistema de sensoriamento de temperatura e umidade de fácil utilização.

REFERÊNCIAS

Lima, Ellen Lima de. **Módulo de sensores para monitoramento da qualidade da água com sem fio utilizando plataforma de prototipagem**. 2018. 79 f. Dissertação (mestrado) - Universidade Estadual de Feira de Santana, Programa de Pós-Graduação em Computação Aplicada, 2018. Acesso em: 08 de ago. 2020.

LIU, Thomas. Digital-output relative humidity & temperature sensor/module. **Sparkfun**. Disponível em: <<https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>>. Acesso em: 06 de jun. 2020.

Adafruit Industries. **DHT-sensor-library**, 2019. Página de postagem de códigos. Disponível em: <<https://github.com/adafruit/DHT-sensor-library>>. Acesso em: 06 de jun. 2020.

OLIVEIRA, Euler. Como usar com Arduino – Sensor de Umidade e Temperatura DHT22 / AM2302. **Blog Master Walker Shop**, 2018. Disponível em: <<https://blogmasterwalkershop.com.br/arduino/como-usar-com-arduino-sensor-de-umidade-e-temperatura-dht22-am2302/>>. Acesso em: 06 de jun. 2020.

MOTA, Allan. DHT11 e DHT22, Sensor de umidade e Temperatura com Arduino. **Portal Vila de Silício**, 2018. Disponível em: <<https://portal.vidadesilicio.com.br/dht11-dht22-sensor-de-umidade-e-temperatura/>>. Acesso em: 06 de jun. 2020.

Microsoft. **Microsoft Power BI Community**, 2020. Tópico de fórum. Disponível em: <<https://community.powerbi.com/t5/Desktop/Select-column-to-show-in-a-graph-card/m-p/1320230>>. Acesso em: 24 ago. 2020.

CURBAL. Change measures using slicers in Power BI Desktop- Power BI Tips & Tricks #20, 2016. (9m13s). Disponível em: <<https://www.youtube.com/watch?v=gYbGNeYD4OY>>. Acesso em: 24 ago. 2020.

QUEVAUVILLIERS, Gilbert. Power BI – Using a Slicer to show different measures. **FourMoo**, 2019. Disponível em: <<https://www.fourmoo.com/2017/11/21/power-bi-using-a-slicer-to-show-different-measures/>>. Acesso em: 24 de ago. 2020.

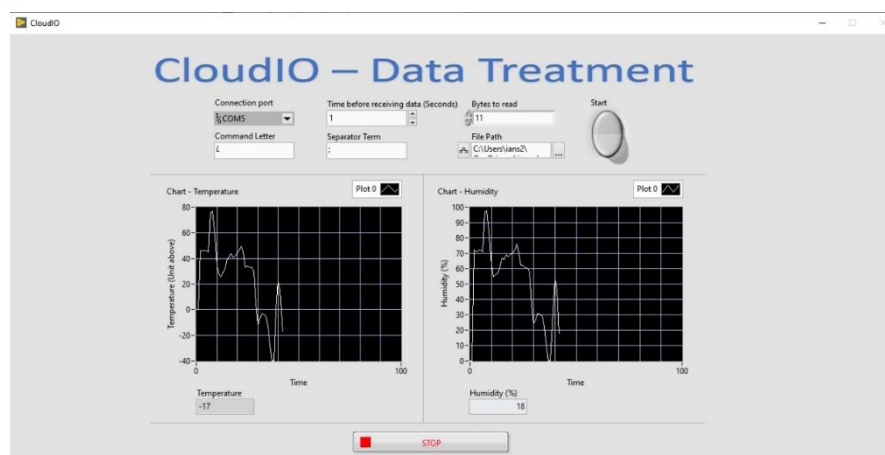
APÊNDICE 1

Código principal

Código principal executado na plataforma do Lab View.

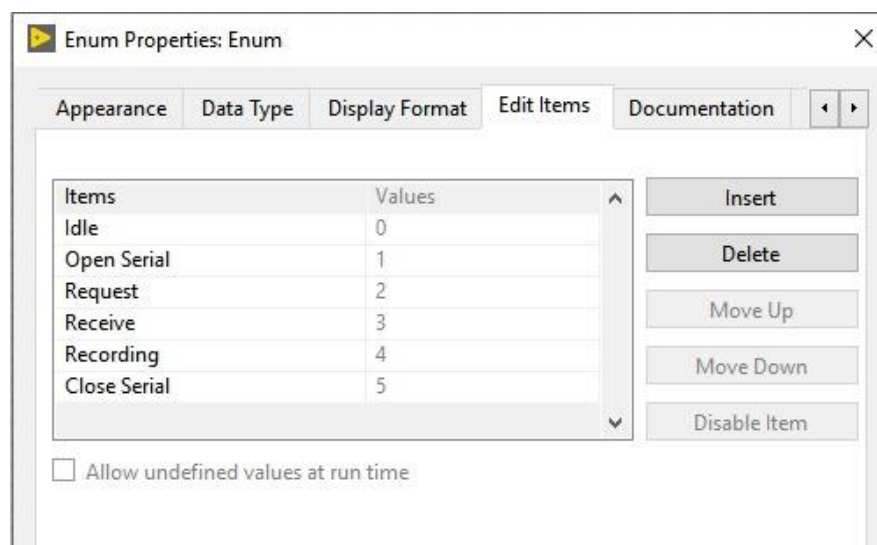
Programa principal:

Figura 4 – Interface do programa no Lab View



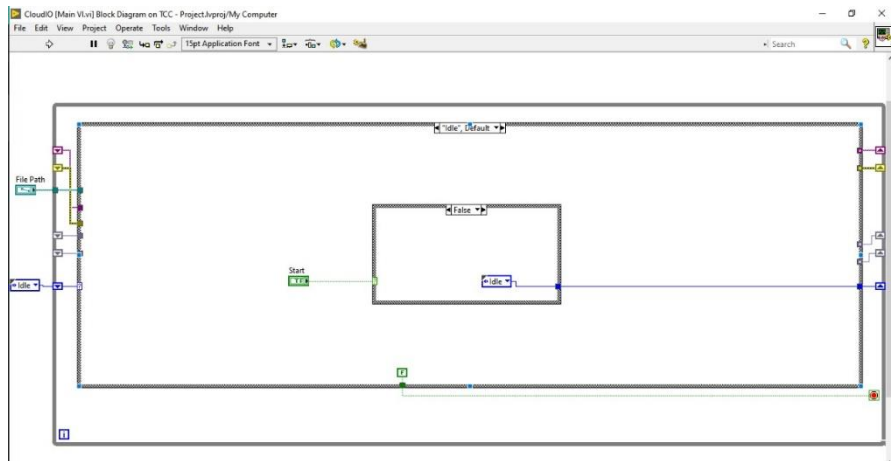
Fonte: Autoria própria, 2020

Figura 5 – Listamento dos estados do programa



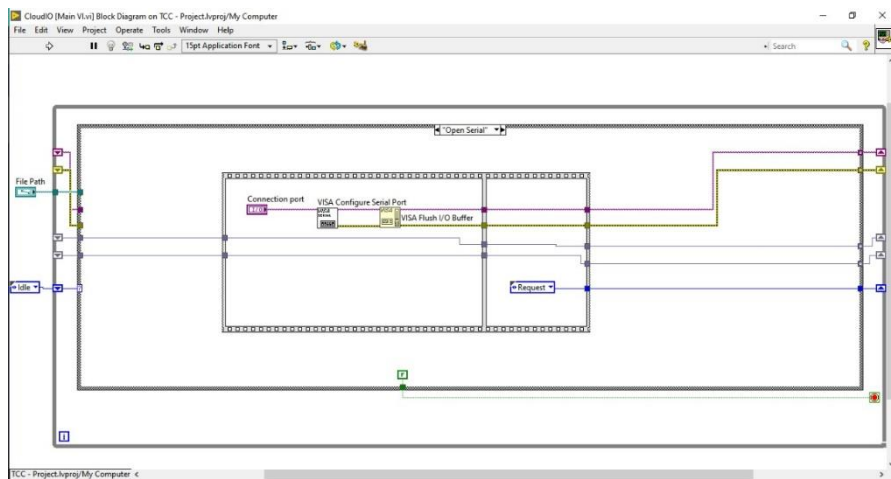
Fonte: Autoria própria, 2020

Figura 6 – Programa em estado “Idle”



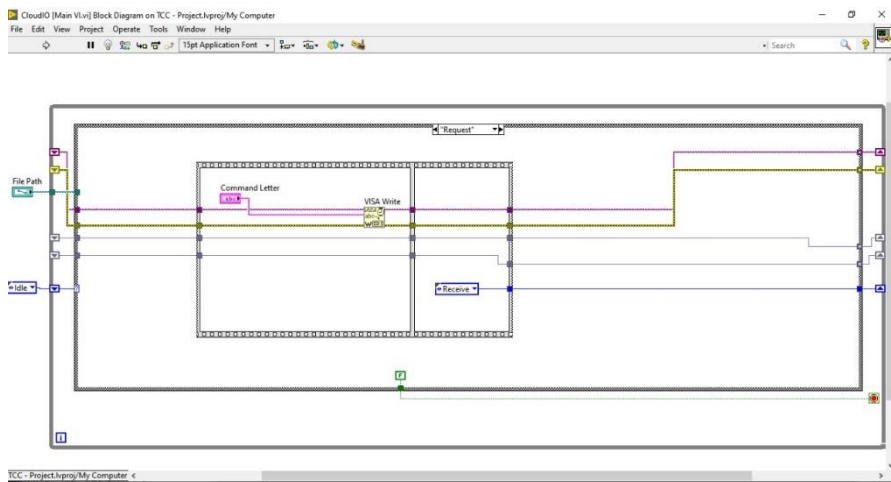
Fonte: Autoria própria, 2020

Figura 7 – Programa em estado “Open serial”



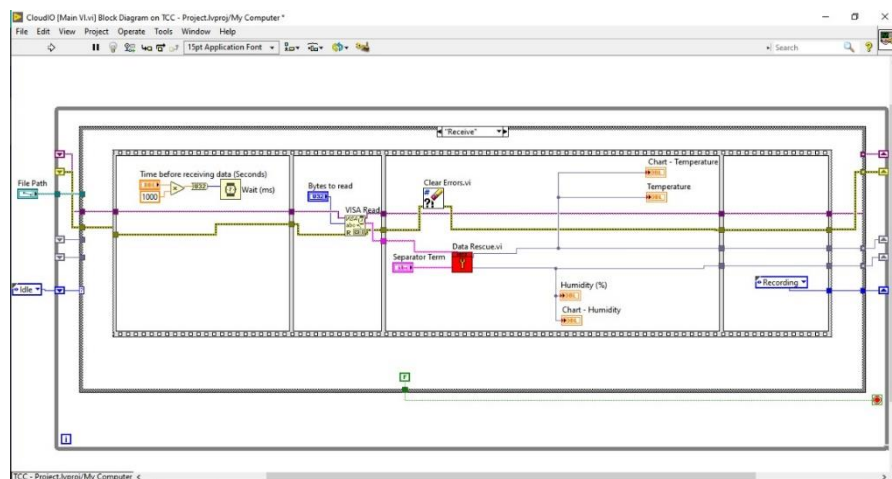
Fonte: Autoria própria, 2020

Figura 8— Programa em estado “Request”



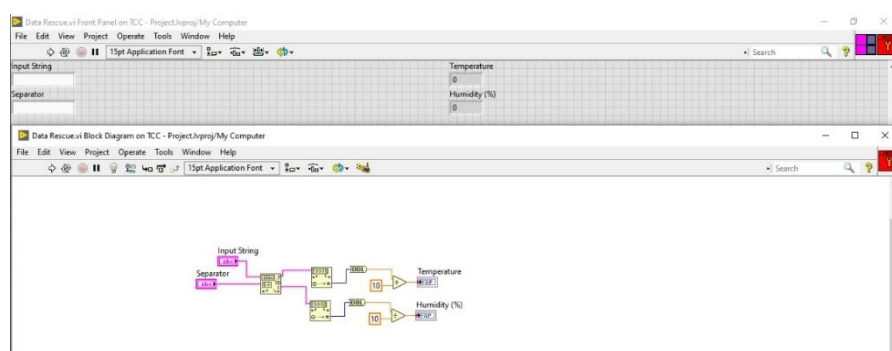
Fonte: Autoria própria, 2020

Figura 9 – Programa em estado “Recieve”



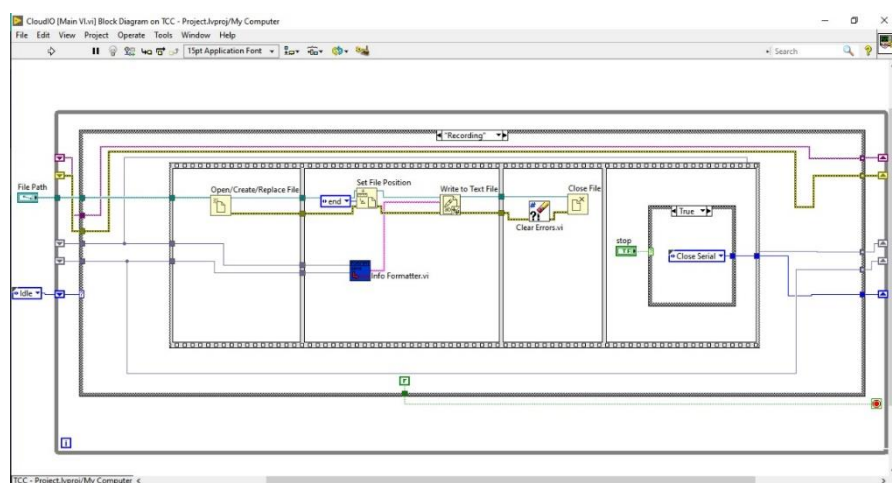
Fonte: Autoria própria, 2020

Figura 10 – Descrição do bloco “Data rescue”



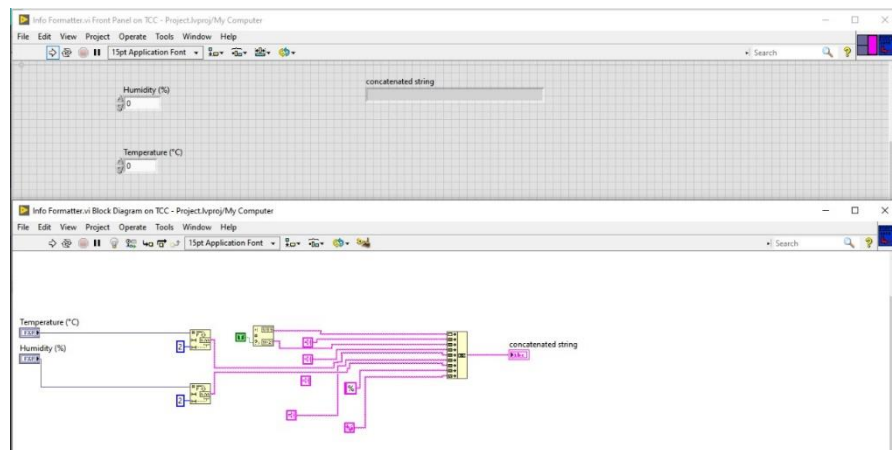
Fonte: Autoria própria, 2020

Figura 11 – Programa em estado “Recording”



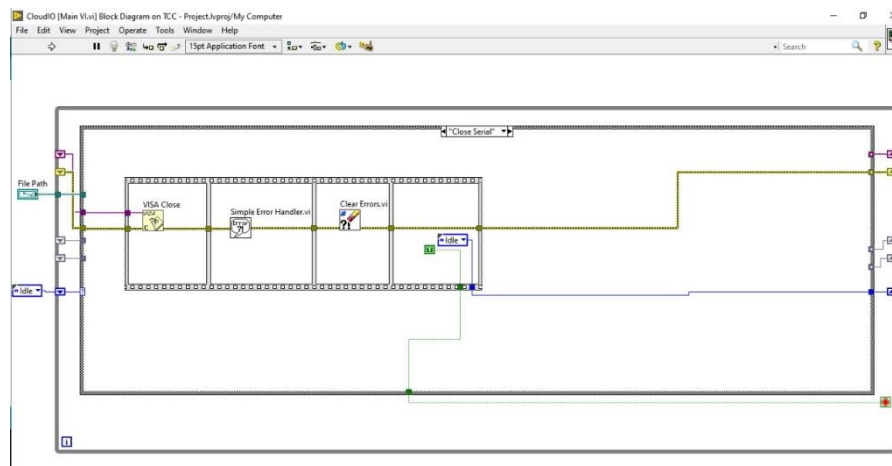
Fonte: Autoria própria, 2020

Figura 12 – Descrição do bloco “Info Formatter”



Fonte: Autoria própria, 2020

Figura 13 – Programa em estado “Close serial”



Fonte: Autoria própria, 2020

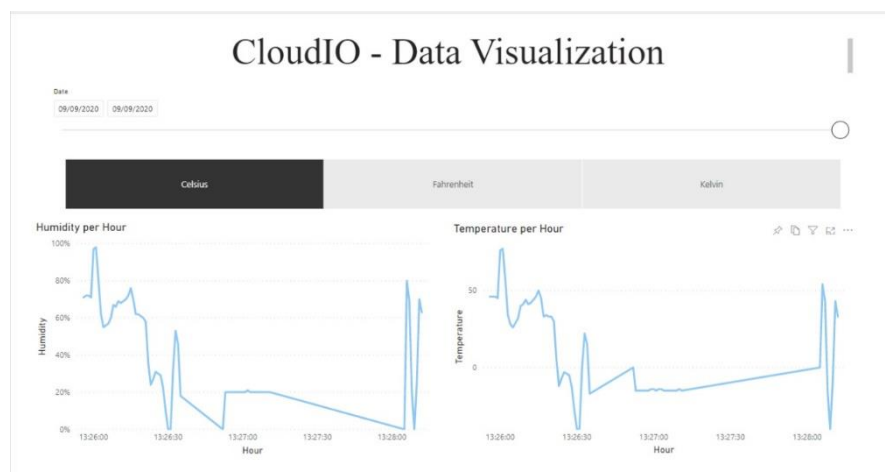
Figura 14 – Arquivo editado pelo programa

Dados Novo.txt - Bloco de Notas

Arquivo	Editar	Formatar	Exibir	Ajuda
09/09/2020		13:26:29	-28,00	10,00%
09/09/2020		13:26:30	-40,00	0,00%
09/09/2020		13:26:31	-40,00	0,00%
09/09/2020		13:26:32	0,00	32,00%
09/09/2020		13:26:33	22,00	53,00%
09/09/2020		13:26:34	15,00	46,00%
09/09/2020		13:26:35	-17,00	18,00%
09/09/2020		13:26:52	0,00	0,00%
09/09/2020		13:26:53	-15,00	20,00%
09/09/2020		13:26:54	-15,00	20,00%
09/09/2020		13:26:55	-15,00	20,00%
09/09/2020		13:26:56	-15,00	20,00%
09/09/2020		13:26:57	-15,00	20,00%
09/09/2020		13:26:58	-15,00	20,00%
09/09/2020		13:26:59	-14,00	20,00%
09/09/2020		13:27:00	-14,00	20,00%
09/09/2020		13:27:01	-15,00	20,00%
09/09/2020		13:27:02	-14,00	21,00%
09/09/2020		13:27:03	-14,00	20,00%
09/09/2020		13:27:04	-15,00	20,00%
09/09/2020		13:27:05	-15,00	20,00%
09/09/2020		13:27:06	-15,00	20,00%
09/09/2020		13:27:07	-15,00	20,00%
09/09/2020		13:27:08	-15,00	20,00%
09/09/2020		13:27:09	-15,00	20,00%
09/09/2020		13:27:10	-14,00	20,00%
09/09/2020		13:27:11	-15,00	20,00%
09/09/2020		13:28:05	0,00	0,00%
09/09/2020		13:28:06	54,00	80,00%
09/09/2020		13:28:07	43,00	69,00%
09/09/2020		13:28:08	-16,00	20,00%
09/09/2020		13:28:09	-40,00	0,00%
09/09/2020		13:28:10	-9,00	25,00%
09/09/2020		13:28:11	43,00	70,00%
09/09/2020		13:28:12	33,00	63,00%

Fonte: Autoria própria, 2020

Figura 15 – Representação gráfica na interface do Power BI



Fonte: Autoria própria, 2020

APÊNDICE 2

Código utilizado no arduino

Código, produzido pelo grupo, utilizado na conversão dos sinais captados pelos sensores (temperatura em umidade) em dados, que depois serão tratados no código exibido no apêndice 1.

Código:

```
/*Course Conclusion Project - CloudIO

Responsible technicians:
  Gustavo Higuti de Souza      - number 11
  Ian Aguila Sanchez          - number 12
  Igor de Lucca Carneiro Antonio - number 13

Description of the code:
  This code is made to read a DHT22 humidity and temperature sensor measurements
  using an Arduino board, to pass this data serially to a computer, for it to
  send the data to the cloud

First version: 1.0 - made at 03:46 AM of 06/06/2020
Previous version : 2.0 - made at 11:49PM of 16/08/2020
Actual version : 3.0 - FINAL - made at 12:21PM of 08/09/2020
*/
//-----Libraries Inclusion -----
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>

//-- Definition of the pin where the device is connected and the type of used sensor --
#define DHTPIN 2    // Digital pin connected to the DHT sensor

// Feather HUZZAH ESP8266 note: use pins 3, 4, 5, 12, 13 or 14 --
// Pin 15 can work but DHT must be disconnected during program upload.
#define DHTTYPE     DHT22

DHT_Unified dht(DHTPIN, DHTTYPE);

//----- Global Variables Declaration -----
int humidity, temperature; // Variables to maintain the temperature and humidity values
multiplied by 10 (to transform in integer values)

char temperatureSignal; // Variable that maintains the temperature signal (if it is
negative or positive)

int hDecimal, hUnit, hTen, hHundred, tDecimal, tUnit, tTen; // Variables that split the
data into each decimal place

String dataToSend = ""; // String that will maintain the whole data concatenated to send
```

```

//-----

void setup() {
  Serial.begin(9600); //Serial communication begin with a baud rate of 9600
  dht.begin(); //Sensor begin
  delay(2000); //Mandatory delay of 2 seconds to start the circuit
}

void loop() {

  if(Serial.available()) // If there is anything at the serial buffer it will read
  {
    if(Serial.read() == 'L') // If the character at the buffer is an L, then the sensor will
    measure the temperature and humidity and the data will be sent through USB
    {

      // Section where the humidity and temperature values are measured, transformed into an
      integer value and attributed to its respective variables
      sensors_event_t event;
      temperature = 10 * (event.temperature);
      dht.humidity().getEvent(&event);
      humidity = 10 * (event.relative_humidity);

      // Section where the signal of the temperature is recorded and its module is obtained and
      attributed to the same variable that it was
      if(temperature < 0)
      {
        temperatureSignal = "-";
        temperature *= -1;
      }
      else
      {
        temperatureSignal = "+";
      }

      //Section where the temperature and humidity data are splitted into decimal places and
      attributed to integer variables
      tTen = temperature / 100;
      tUnit = temperature / 10 - (10 * tTen);
      tDecimal = temperature - (10 * tUnit) - (100 * tTen);

      hHundred = humidity / 1000;
      hTen = humidity / 100 - (10 * hHundred);
      hUnit = humidity / 10 - (100 * hHundred) - (10 * hTen);
      hDecimal = humidity - (1000 * hHundred) - (100 * hTen) - (10 * hUnit);

      //----- Section where the data will be concatenated to be sent -----
      dataToSend.concat(temperatureSignal);
      dataToSend.concat(tTen);
      dataToSend.concat(tUnit);
      dataToSend.concat(tDecimal);
      dataToSend.concat(';');
      dataToSend.concat(hHundred);
      dataToSend.concat(hTen);
      dataToSend.concat(hUnit);
      dataToSend.concat(hDecimal);

      //-----
      Serial.println(dataToSend); // Serial transmission of the data

    }
    Serial.flush(); // If there was anything at the serial buffer, wether it was an L or
    not, the buffer will be flushed
  }
}

```