INSTITUTO FEDERAL DE SANTA CATARINA

Daniel Wzoreck

Otimização de Pequenos Processos na Agricultura Regional com a Tecnologia da Informação

Canoinhas – SC
09/10/2019

INSTITUTO FEDERAL DE SANTA CATARINA

Daniel Wzoreck

Otimização de Pequenos Processos na Agricultura Regional com a Tecnologia da Informação

Relatório apresentado ao curso Montagem e Manutenção - câmpus Canoinhas do Instituto Federal de Santa Catarina para a obtenção do diploma de Técnico.

Orientador: Diocelio Larsen

Canoinhas - SC

12/2019

RESUMO

Com o intuito de solucionar alguma necessidade real existente na sociedade, nos deparamos com a estufa do IFSC Câmpus Canoinhas, um local cuja única tecnologia implementada é um sistema de irrigação que atua de forma mecânica, tendo em vista que é um excelente local onde pode-se gerar análise de dados, implementamos um sistema capaz de coletar alguns dados e automatizar o processo de irrigação, utilizando um microcomputador e dois sensores, este microcomputador é o responsável por receber os dados vindos dos sensores e enviá-los para uma base de dados presente no laboratório de redes, desenvolvemos um site para apresentar estes dados, porém ainda resta realizar a programação *backend* responsável pela comunicação entre a base de dados e o site. Durante uma semana mais ou menos o projeto ficou coletando dados de umidade e temperatura da estufa, foram gerados cerca de 11.000 dados, foi possível analisar a eficiência da estufa em relação ao ambiente externo, nos dias mais frios se mantendo cerca de 8 graus acima da temperatura externa e em dias quentes cerca de 20 graus acima. Ocorreram dois pequenos incidentes o sensor externo foi danificado por questões climáticas, suas leituras de umidade não são mais precisas, mas as leituras de temperatura ainda se mostram confiáveis, o outro problema é a questão do superaquecimento do microcomputador por estar dentro de uma caixa onde não há nenhuma espécie de refrigeração, quando faz muito calor ele tende a ficar um certo tempo desligado para evitar danos ao hardware. De resto o projeto se mostrou muito eficiente e funcional.

ABSTRACT

In order to solve any real need in society, we came across the greenhouse of IFSC Câmpus Canoinhas, a place whose only implemented technology is a mechanically acting irrigation system, as it is an excellent place where If we generate data analysis, we implemented a system capable of collecting some data and automating the irrigation process, using a microcomputer and two sensors, this microcomputer is responsible for receiving the data coming from the sensors and sending it to a present database. In the network lab, we developed a website to present this data, but we still have to do the backend programming responsible for the communication between the database and the site. For a week or so the project was collecting greenhouse humidity and temperature data, about 11,000 data were generated, it was possible to analyze the efficiency of the greenhouse in relation to the external environment, on colder days staying about 8 degrees above outside temperature and on hot days about 20 degrees above. Two minor incidents have occurred the external sensor has been damaged by weather, its humidity readings are no longer accurate, but the temperature readings are still reliable, the other problem is the microcomputer overheating due to being inside a box where it is not. There is no kind of cooling, when it gets too hot it tends to be off for a while to avoid hardware damage. Moreover the project proved to be very efficient and functional.

LISTA DE FIGURAS

Figura 1: Sensor DHT11 e sua pinagem.....13
Figura 2: Ligação DHT11 no Raspberry Pi 3.....14
Figura 3: DHT11 com os resistores já soldados.....14
Figura 4: Sensores já conectados no Raspberry Pi 3.....15
Figura 5: Ligação nos pinos GPIO do Raspberri Pi 3.....15
Figura 6: Primeiras leituras dos sensores.....17
Figura 7: Carcaça de antena antiga usada para proteger o hardware.....23
Figura 8: Aproveitamento de conectores RJ45 para realizar a comunicação com os sensores.....24
Figura 9: Fixação do Raspberry no suporte interno da antena.....24
Figura 10: Instalação final na estufa.....25
Figura 11: Sensor interno da estufa.....25
Figura 12: Página de monitoramento do site.....26
Figura 13: Página de histórico do site.....27
Figura 14: Página de histórico 2.....28
Figura 15: Página apresentado o projeto.....28
Figura 16: Registros no Banco de Dados.....29

LISTA DE ABREVIATURAS E SIGLAS

IFSC – Instituto Federal de Santa Catarina;

PHP – Hypertext Preprocessor (uma linguagem de programação);

SGBD – Sistema de Gerenciamento de Banco de Dados;

BD – Banco de Dados;

Web – Palavra inglesa que significa teia ou rede. O significado de *web* ganhou um novo sentido com o aparecimento da internet, passou a designar a rede que conecta computadores;

TI – Tecnologia de Informação;

GPIO – General Purpose Input/Output (Entrada e Saída de Propósito Geral);

SMD – Surface Mount Technology (Tecnologia de montagem em superfície).

SUMÁRIO

1 INTRODUÇÃO....8

1.1 Objetivos....9
1.1.1 Objetivo geral....9
1.1.2 Objetivo específico....9
2 DESENVOLVIMENTO....10

2.1 Revisão de literatura....10
3 METODOLOGIA....12

3.1 Sensor DHT11....12
3.1.1 Ligação no Raspberry Pi....12
3.2 Teste dos sensores....14
3.3 Instalação do SGBD MySQL no Linux (Ubuntu ou derivado) e criação de usuário remoto....16
3.3.1 Instalação do MySQL Server....17
3.3.2 Configurações necessárias....17
Privilégios do usuário....18

Privilégios do usuário....18
Atualizar a tabela de privilégios....18

Atualizar a tabela de privilégios....18
3.4 Criação do banco e tabela....19
3.8 Site....25

4 ANÁLISE E DISCUSSÃO DOS RESULTADOS....28

4.1 Funcionamento esperado dos sensores....29
4.2 Problemas de aquecimento no Raspberry....29
5 CONCLUSÃO....30

REFERÊNCIAS....31

1 INTRODUÇÃO

Atuar no processo de desenvolvimento de um software que venha a contribuir com algo ou alguém, é motivante, juntamente a todo aprendizado e conhecimento adquirido no decorrer do processo. Partindo do zero, onde há somente uma ideia, até chegar ao ponto de ter um “produto” que seja funcional e atenda um grupo específico de pessoas.

O problema surge do sistema de irrigação atual presente na estufa, que atua de forma mecânica, onde o trabalho está sempre em configurar e “programar” o *timer* para ligar a bomba de irrigação em horários específicos. Isso acaba gerando um certo comprometimento, preocupação e desconforto, pois caso o timer seja programado para ligar em certos períodos, e no decorrer do dia o clima acabe variando, é necessário presencialmente desligar de forma manual o sistema de irrigação, ou em finais de semana e feriados, onde não há pessoas para estar cuidando e programando o sistema. Isso pode prejudicar ou até mesmo matar os cultivos que estão dentro da estufa. Outro ponto interessante é o fato da estufa ser um excelente local que pode a vir gerar diversos dados favoráveis ao meio acadêmico, científico e tecnológico, pois ainda não há tecnologia implementada para este fim no local.

O objetivo é desenvolver um sistema que atue de forma automática no processo de acionamento da irrigação a partir de sensores. Juntamente a isso aproveitando as leituras dos sensores para gerar uma base de dados, o usuário terá acesso a estes através de uma plataforma Web que será desenvolvida especificamente para favorecer a análise, estudo e compreensão de tais dados, assim podendo gerar gráficos, consultas, planilhas, estatísticas e outras inúmeras possibilidades que podem ser implementadas durante a vida do software. Esperamos que nosso projeto venha a vir contribuir para o cenário cientifico no IFSC câmpus Canoinhas e região.

No primeiro momento, durante o segundo semestre de 2019 no período do projeto integrador “PI”, será realizado testes e estudos para verificar a viabilidade do projeto em questões tecnológicas e humanas, assim garantindo que os futuros materiais a serem adquiridos estejam de acordo com o objetivo do projeto, e que seja adquirido somente o necessário, assim estendendo o desenvolvimento durante o ano de 2020, o qual esperasse concluir todos os objetivos citados. Esta ideia foi inscrita no Edital PROEX

número 16/2019 (Protagonismo Discente), sendo aprovado e contemplado com uma bolsa de R$3.400,00, cuja bolsa a ser aplicada no projeto e sem a necessidade de prestação de contas, o edital prevê até 10/03/2020 para envio do relatório final, mas a equipe se disponibiliza estender até o final do ano de 2020 para a conclusão do projeto, a primeiro momento até 10/03/2020 pretendemos realizar toda a parte que ficará instalada na estufa, como sensores, componentes e o microcontrolador responsável pela coleta de dados, controle de algumas situações e envio de leituras para a base de dados, após esta etapa realizaremos todo o desenvolvimento do site, que será a parte voltada ao usuário final.

1.1 Objetivos

1.1.1 Objetivo geral

Desenvolver um sistema de irrigação automática e armazenamento de dados, utilizando sensores e placas microcontroladoras.

1.1.2 Objetivo específico

- Efetuar a aquisição dos equipamentos necessários;
- Configurar e instalar um Raspberry Pi na estufa;
- Desenvolver um algoritmo em python que rode no Raspberry e seja capaz de coletar dados e enviar para o banco de dados;
- Buscar soluções para evitar que o hardware externo fique exposto à chuva;
- Implementar banco de dados SQL com acesso remoto;
- Desenvolver o site o qual se comunicará com o banco de dados.

2 DESENVOLVIMENTO

Tendo definido como objetivo desenvolver um sistema capaz de coletar dados e automatizar o processo de irrigação, partimos em direção a buscas sobre o assunto, com o propósito de familiarização com termos e tecnologias que nos proporcionasse chegar ao objetivo esperado. Inicialmente, realizou-se uma entrevista com os responsáveis pela estufa, na qual foi levantado questões referentes a necessidades tecnológicas e eventuais soluções que pudéssemos desenvolver, o principal ponto requerido foi a questão da irrigação.

2.1 Revisão de literatura

Apesar de haver diversos projetos similares a este, se encontra pouco material publicado, geralmente os projetos são realizados e não há nenhum documento, monografia ou relatório referente a resultados, formas de aplicação entre outros. Levando isso em conta será citado algumas fontes que se aplicam ao escopo deste projeto.

Primeiramente um conceito em alta e que se aplica ao projeto é o de internet das coisas, ou seja:

> a internet das coisas nada mais é que uma rede de objetos físicos (veículos, prédios e outros dotados de tecnologia embarcada, sensores e conexão com a rede) capaz de reunir e de transmitir dados. É uma extensão da internet atual que possibilita que objetos do dia a dia, quaisquer que sejam mas que tenham capacidade computacional e de comunicação, se conectem à Internet. A conexão com a rede mundial de computadores possibilita, em primeiro lugar, controlar remotamente os objetos e, em segundo lugar, que os próprios objetos sejam usados como provedores de serviços. (WIKIPÉDIA, 2019)

Outro conceito importantíssimo na essência do projeto é o de agricultura de precisão, ou seja:

“Agricultura de precisão está associado à utilização de aparelhagem de tecnologia avançada para avaliar e acompanhar de maneira mais precisa as condições das áreas de atividades agronômicas baseada no princípio da variabilidade do solo e clima.” (WIKIPÉDIA, 2019)

3 METODOLOGIA

Utilizamos um Raspberry Pi 3 modelo B como micro computador responsável por acionar os sensores, receber as informações e enviá-las para o banco de dados presente no laboratório de Redes. Para capturar os dados foi utilizado dois sensores DHT11, cada um realiza a leitura de umidade do ar e temperatura ambiente, para realizar a captura e envio de dados utilizamos a linguagem de programação *Python* no Raspberry Pi.

3.1 Sensor DHT11



Figura 1: Sensor DHT11 e sua pinagem

Características:

- Alimentação: 3 à 5,5 V;
- Faixa de leitura – Umidade: 20 à 80%;
- Precisão umidade: 5%;
- Faixa de leitura – Temperatura: 0 – 50 ºC;
- Precisão temperatura: +/- 2 ºC.

3.1.1 Ligação no Raspberry Pi

O DHT11 necessita de um resistor de 4.7KΩ entre o pino 1 e 2 como pull-up, para seu funcionamento com o barramento GPIO do Raspberry devemos utilizar os pinos 1, 2 e 4 (VCC, Dados e GND) do sensor, o pino 1 do sensor deve ser ligado na conexão 3.3V

da placa, o pino 2 em um pino digital qualquer, no nosso caso utilizamos o pino BCM 4 e por fim o pino 4 deve ser ligado na conexão GND, para saber mais sobre a pinagem do Raspberry Pi 3 acesse: https://pinout.xyz/#.

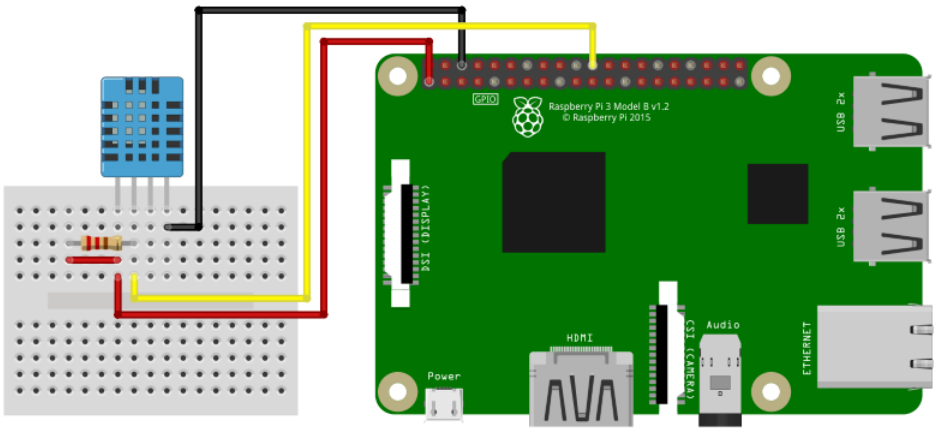


Figura 2: Ligação DHT11 no Raspberry Pi 3

Para realizar a ligação dos dois sensores simultâneos deve-se utilizar o mesmo VCC e GND, bastando apenas adicionar uma nova conexão BCM em um pino digital, no nosso caso foram utilizadas as conexões BCM 4 e BCM 17. Nossa ligação eletrônica ficou da seguinte forma:

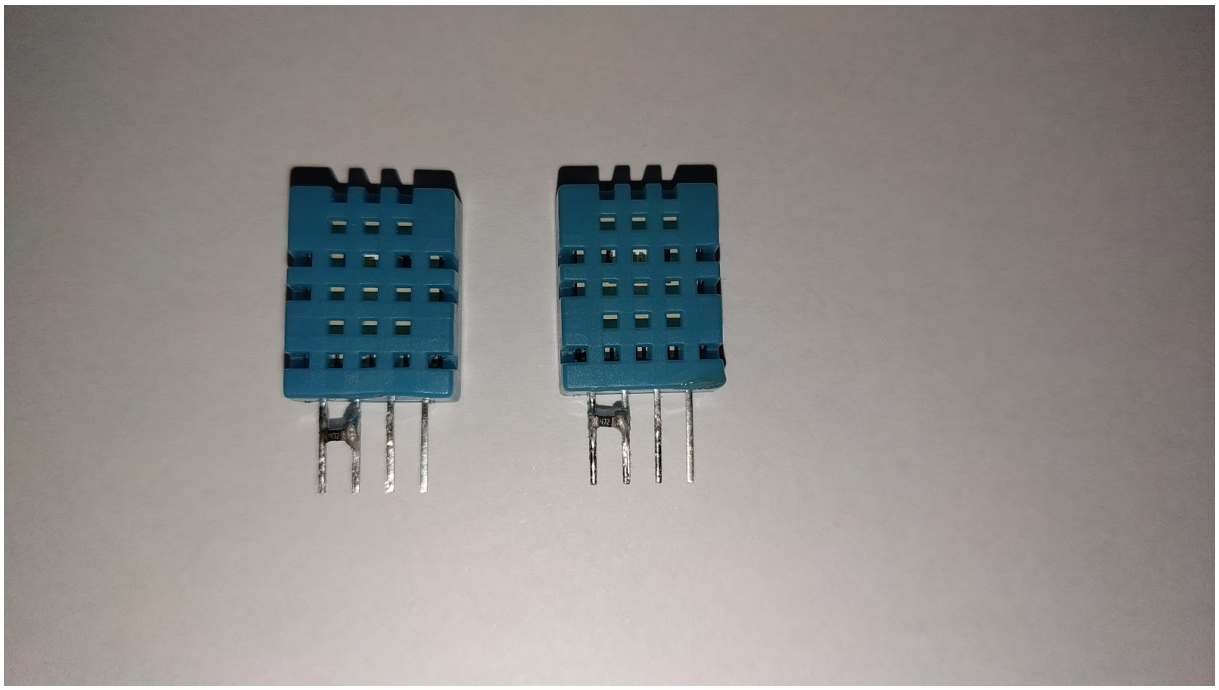


Figura 3: DHT11 com os resistores já soldados

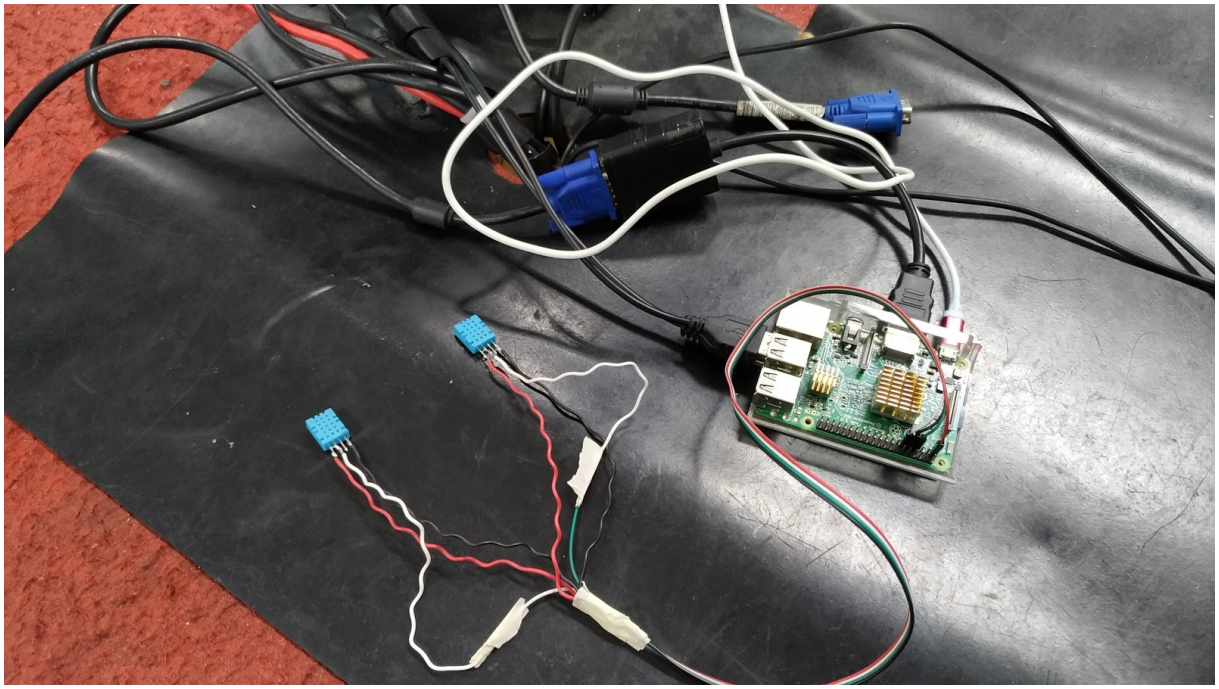


Figura 4: Sensores já conectados no Raspberry Pi 3

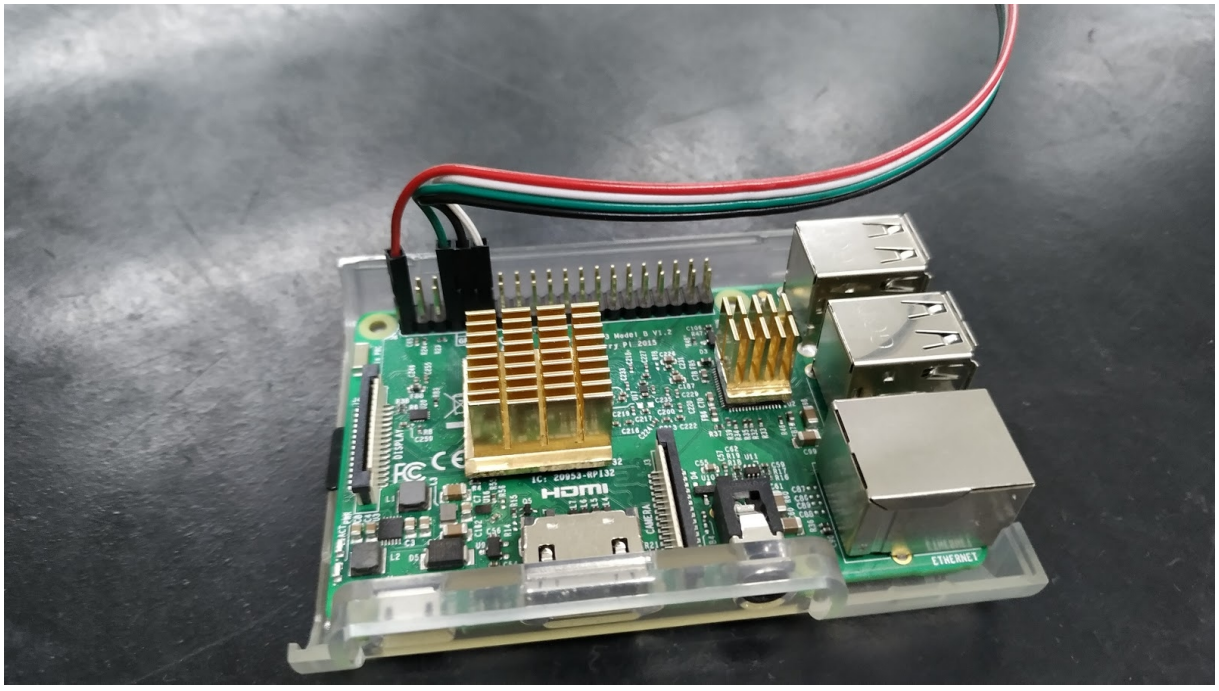


Figura 5: Ligação nos pinos GPIO do Raspberri Pi 3

3.2 Teste dos sensores

Com o sistema operacional Raspbian já instalado e rodando no cartão de memória do Raspberry Pi, devemos instalar a biblioteca **Adafruit_DHT** para o funcionamento dos sensores com a linguagem *Python*, no terminal insira os seguintes comandos:

sudo apt update

sudo apt install build-essential python-dev python-openssl

git clone https://github.com/adafruit/Adafruit_Python_DHT

cd Adafruit_Python_DHT

sudo python setup.py install

sudo python3 setup.py install

Agora basta basta você criar um arquivo com extensão .py e adicionar o seguinte código:

```
import Adafruit_DHT as dht
import time as t

while True:
      umidade_interna, temperatura_interna = dht.read_retry(dht.DHT11, 4)
      t.sleep(2)
      umidade_externa, temperatura_externa = dht.read_retry(dht.DHT11, 17)
      print('\n\nTemperatura      interna:      {0:0.1f}      Umidade      interna
{1:0.1f}'.format(temperatura_interna, umidade_interna))
      print('\nTemperatura      externa:      {0:0.1f}      Umidade      externa:
{1:0.1f}'.format(temperatura_externa, umidade_externa))
      t.sleep(2)
```

Salve o programa, entre no terminal e vá até o diretório onde o programa se encontra, então execute o seguinte comando: python nome_do_seu_programa.py , se tudo ocorreu corretamente você começará a visualizar a leitura dos sensores.

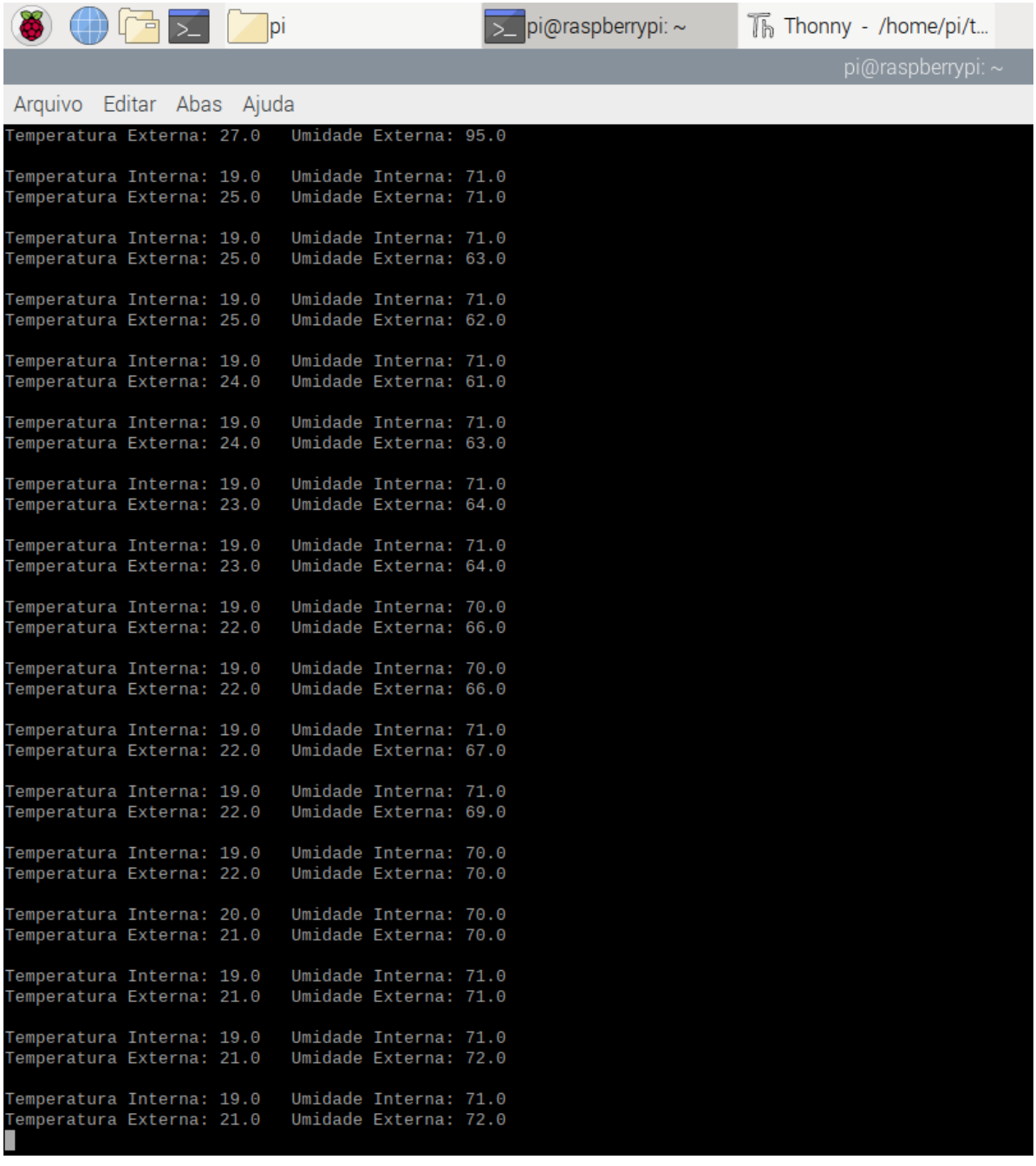


Figura 6: Primeiras leituras dos sensores

3.3 Instalação do SGBD MySQL no Linux (Ubuntu ou derivado) e criação de usuário remoto

Primeiramente devemos entrar no terminal, basta usar `ctrl+alt+t` ou procurar ele no seu menu de aplicativos. Para realizar a instalação você deve copiar os comandos a partir do `$` e colar em seu terminal, para colar no terminal utilize `ctrl+shift+v`.

3.3.1 Instalação do MySQL Server

Execute o seguinte comando: usuario@usuario:~$ sudo apt install mysql-server

Será pedido a sua senha de **root** para poder realizar a instalação, basta informá-la e a instalação prosseguirá normalmente, caso durante o processo seja solicitado algo como "Deseja prosseguir a instalação? [S/n]" ou "Você quer continuar? [S/n]" basta informar S e pressionar enter.

3.3.2 Configurações necessárias

Com o comando abaixo iremos solicitar o processo de configuração inicial do MySQL, basta criar uma senha de **root** e responder umas perguntas, caso não tenha conhecimento responda não para todas.

usuario@usuario:~$ sudo mysql_secure_installation

3.3.3 Permitir conexão remota

No documento abaixo procure a linha **bind-address** e mude o IP dela para **0.0.0.0**, após isso pressione ctrl + x e enter .

usuario@usuario:~$ sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf

Devemos liberar a porta 3306 com o seguinte comando:
usuario@usuario:~$ sudo ufw allow 3306

Agora você precisa reiniciar o serviço, execute o seguinte comando: usuario@usuario:~$ sudo service mysql restart

3.3.4 Configurações no SGBD

Primeiramente devemos entrar no SGBD, use o seguinte comando: `sudo mysql -u root -p`

Você pode alterar sua senha de root, lembre-se ela deve ser uma senha segura, o usuário **root** é o que tem os mais alto privilégio no SGBD, caso queira alterar sua senha de **root** use:

```
ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'informe_sua_senha_aqui';
```

3.3.5 Criação de usuário com permissão de acesso remoto

A criação deste usuário será com todos os privilégios, o que não é aconselhável de se fazer, mas por fins didáticos criaremos um usuário com todos os privilégios.

Usuário no *localhost* e de um IP qualquer

```
CREATE USER 'nome_do_seu_usuário'@'localhost' IDENTIFIED BY 'sua_senha';
```

```
CREATE USER 'nome_do_seu_usuário'@'%.%.%.%' IDENTIFIED BY 'sua_senha';
```

Privilégios do usuário

```
GRANT ALL PRIVILEGES ON * . * TO 'nome_do_seu_usuário'@'localhost';
```

```
GRANT ALL PRIVILEGES ON * . * TO 'nome_do_seu_usuário'@'%.%.%.%';
```

Atualizar a tabela de privilégios

```
FLUSH PRIVILEGES;
```

3.4 Criação do banco e tabela

Ainda dentro do MySQL com seu usuário, devemos criar um banco de dados primeiramente, basta apenas executar: `CREATE DATABASE estufa;`

Para criar a tabela responsável por armazenar os dados capturados pelos sensores, basta executar:

```
USE estufa;

CREATE TABLE registros(
temp_interna tinyint(2),
temp_externa tinyint(2),
umid_interna tinyint(3),
umid_externa tinyint(3),
dia datetime);
```

3.5 Código final *Python*

Teremos que instalar a biblioteca **pymysql** para podermos gerar a conexão com o BD, no terminal do seu Raspberry Pi use os seguintes comandos:

```
usuario@usuario:~$ sudo apt install python-pip

usuario@usuario:~$ pip install PyMySQL
```

Tendo uma conexão de internet local e estável no Raspberry Pi, o projeto já está pronto para enviar os dados para o BD. Basta criarmos o código final que enviará os dados, no código a seguir na parte **`conexao = pymysql.connect ( )`** você precisa alterar

conforme os seus dados, em **host** informe o IP do seu servidor, **user** nome do seu usuário do MySQL e em **passwd** a senha do seu usuário, crie um arquivo **.py** e insira o seguinte código:

```
import Adafruit_DHT as dht

import time

import pymysql


conexao = pymysql.connect (

    host = 'IP_do_seu_MySQL_Server',

    user = ‘Nome_do_seu_usuário',

    passwd = ‘Senha_do_seu_usuário',

    database = 'estufa'

)


umidade_interna, temperatura_interna = dht.read_retry(dht.DHT11, 4)

time.sleep(2)

umidade_externa, temperatura_externa = dht.read_retry(dht.DHT11, 17)

cursor = conexao.cursor()

cursor.execute("INSERT INTO registros(temp_interna, temp_externa, umid_interna, umid_externa, dia) VALUES ({}, {}, {}, {}, now())".format(temperatura_interna, temperatura_externa, umidade_interna, umidade_externa))
```

```
conexao.commit()

time.sleep(5)
```

Executando no terminal você poderá testar se o código funcionou, execute:

```
usuario@usuario:~$ python nome_do_seu_arquivo.py
```

Se tudo ocorrer corretamente sem erros a leitura dos dados estarão no servidor, no servidor MySQL logado com o seu usuário execute:

```
USE estufa;

SELECT * FROM registros;
```

Você verá um resultado similar a este:

```
|------------------+------------------+-----------------+------------------+----------------------------|
| temp_interna | temp_externa | umid_interna | umid_externa | dia                        |
|------------------+------------------+-----------------+------------------+----------------------------|
|               20 |               28 |              93 |               22 | 2019-10-07 13:57:25 |
|------------------+------------------+-----------------+------------------+----------------------------|
```

3.6 Agendar a execução do programa automaticamente

Como você pode ter percebido a execução do programa anterior ocorreu quando realizado manualmente no terminal, para isto ocorrer automaticamente utilizaremos o ***Crontab*** trata-se de um agendador de tarefas presente no Linux, no nosso caso agendamos a execução do programa para ocorrer a cada minuto, primeiramente copie o caminho do diretório onde se encontra o seu programa, no terminal do seu Raspberry execute o seguinte comando:

```
usuario@usuario:~$ sudo nano /etc/crontab
```

Insira a seguinte linha ao final do arquivo:

```
*/1 *    * * * root       /usr/bin/python /Diretório_do_se_arquivo.py
```

Salve e saia do arquivo, se tudo ocorreu corretamente seu programa já estará rodando e enviando dados para o BD a cada minuto.

3.7 Instalação na estufa

Após todo o procedimento anterior concluído, tivemos que achar uma solução para evitar que o hardware ficasse exposto à chuva e umidade, utilizamos carcaças de antenas via rádio antigas, dentro delas encontramos o seu hardware responsável por receber e enviar dados, o mais interessante para nosso caso é a carcaça pois ela é feita para deixar este hardware exposto ao tempo, então é ideal para montarmos nosso projeto nela.

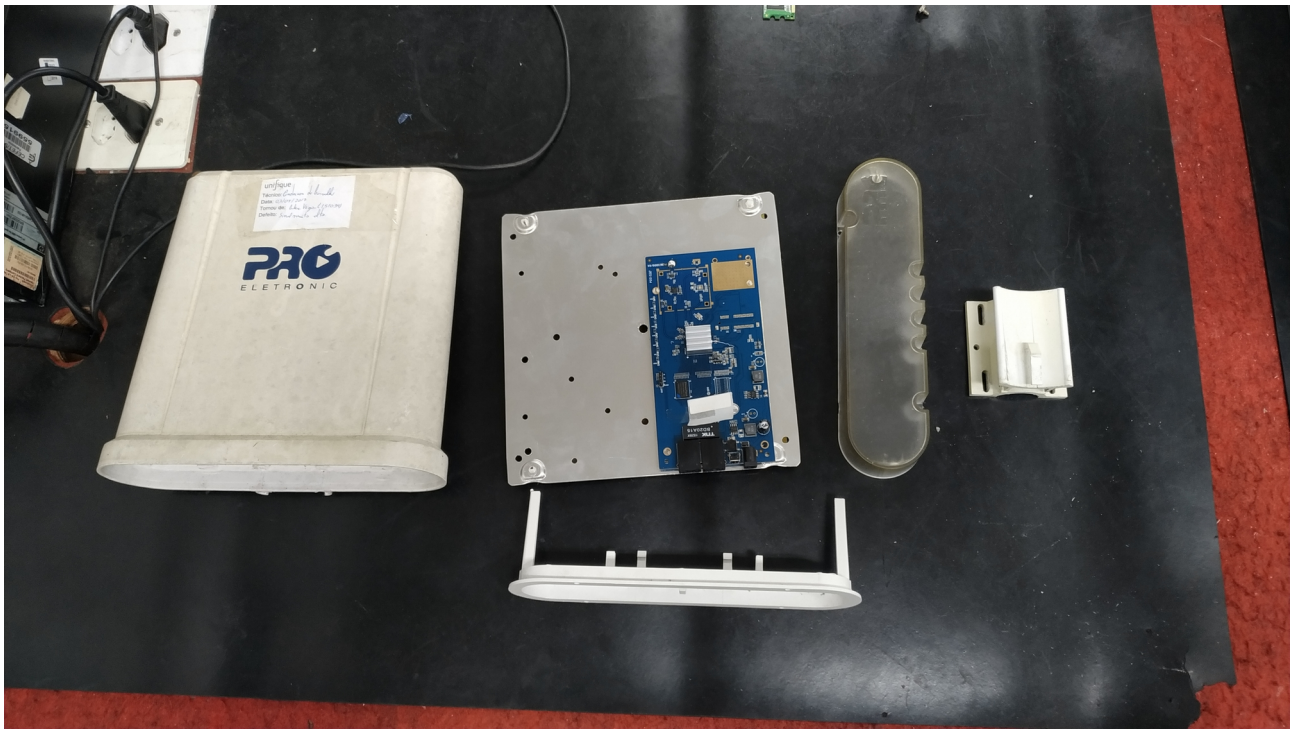


Figura 7: Carcaça de antena antiga usada para proteger o hardware

Foi aproveitado os conectores RJ45 fêmeas da placa original, com eles temos a possibilidade de soldar os pinos que fazem conexão entre o Raspberry e os sensores, assim há a possibilidade de retirar os sensores para eventuais trocas sem a necessidade de abrir a carcaça da antena toda vez. A imagem abaixo mostra os fios que foram

soldados nesses conectores.

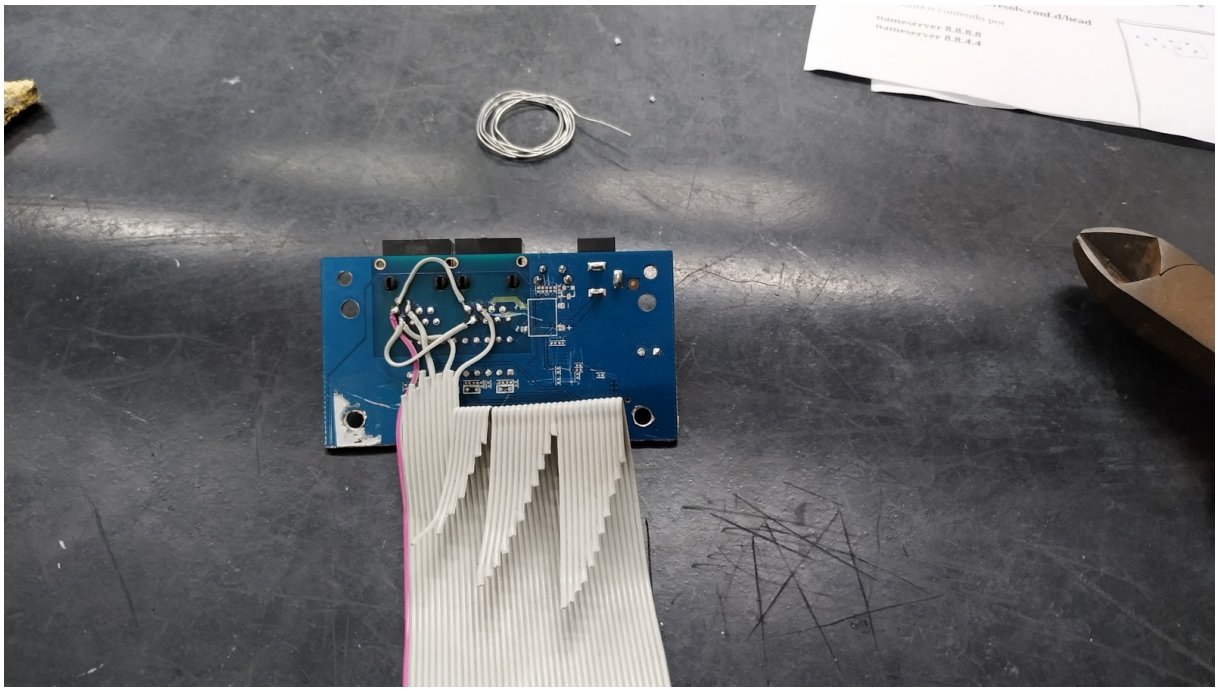


Figura 8: Aproveitamento de conectores RJ45 para realizar a comunicação com os sensores

Como explicado anteriormente a montagem do Raspberry no suporte interno da carcaça fica da seguinte forma.

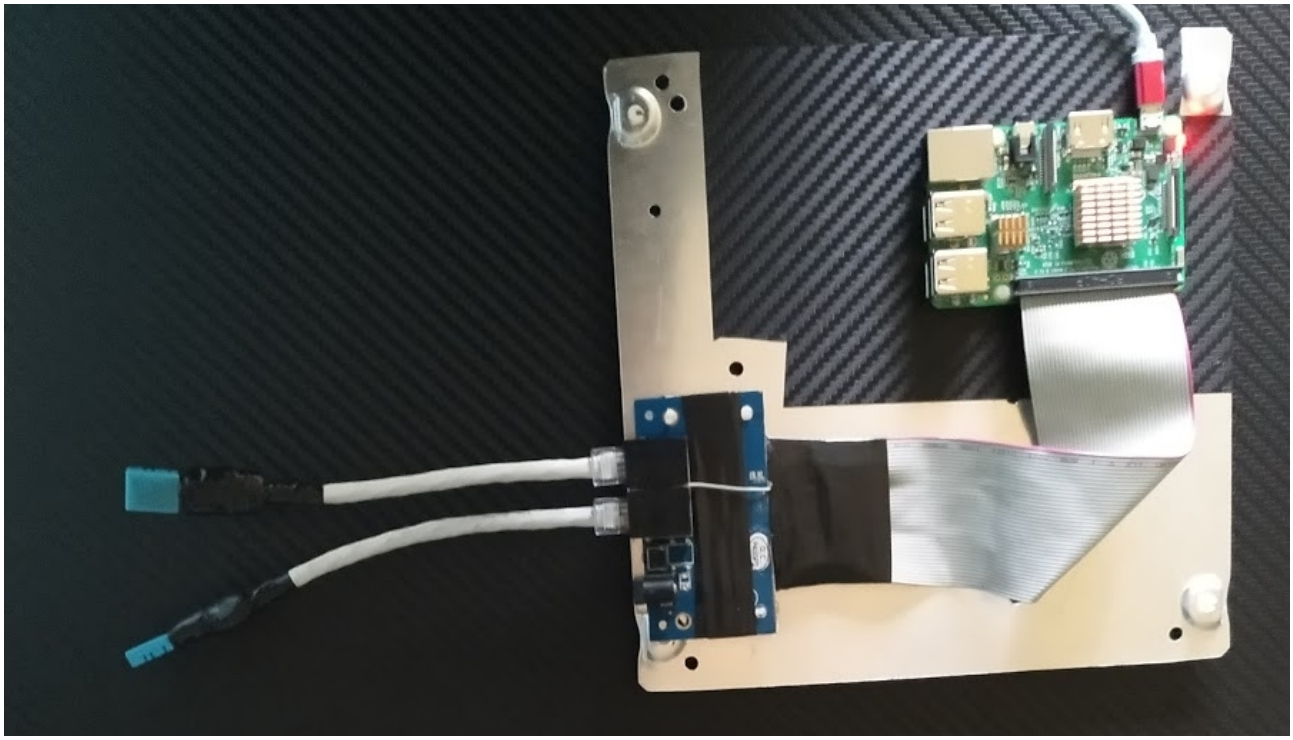


Figura 9: Fixação do Raspberry no suporte interno da antena

Após termos concluído todo o processo, por fim foi instalado o projeto na estufa, o hardware ficou para o lado de fora, próximo a uma tomada instalada no local, neste equipamento externo ficou um sensor DHT11 medindo a temperatura e umidade externa, e já internamente na estufa foi instalado o outro sensor, assim futuramente com uma boa base de dados pode-se gerar algumas comparações relevantes.



Figura 10: Instalação final na estufa



Figura 11: Sensor interno da estufa

3.8 Site

Foi desenvolvido utilizando HTML e CSS, resta realizar a programação backend para possibilitar a comunicação entre o site e a base de dados, o layout ficou da seguinte forma.

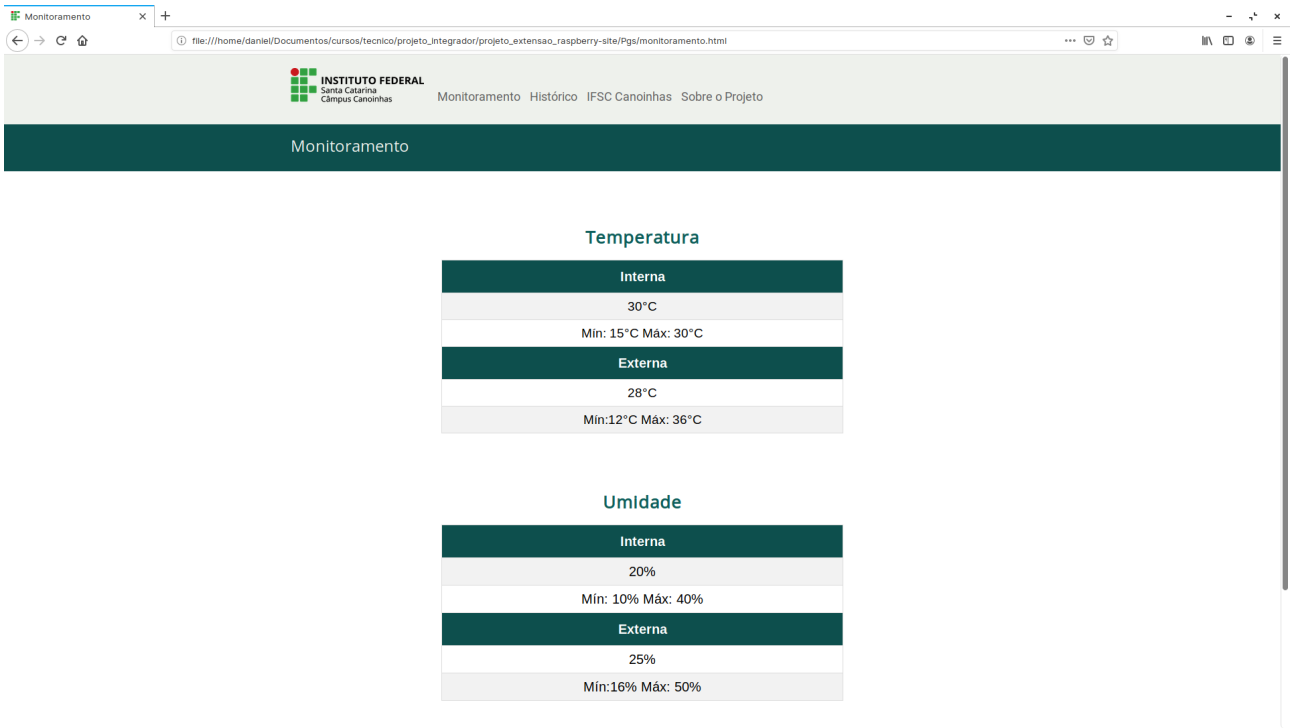


Figura 12: Página de monitoramento do site

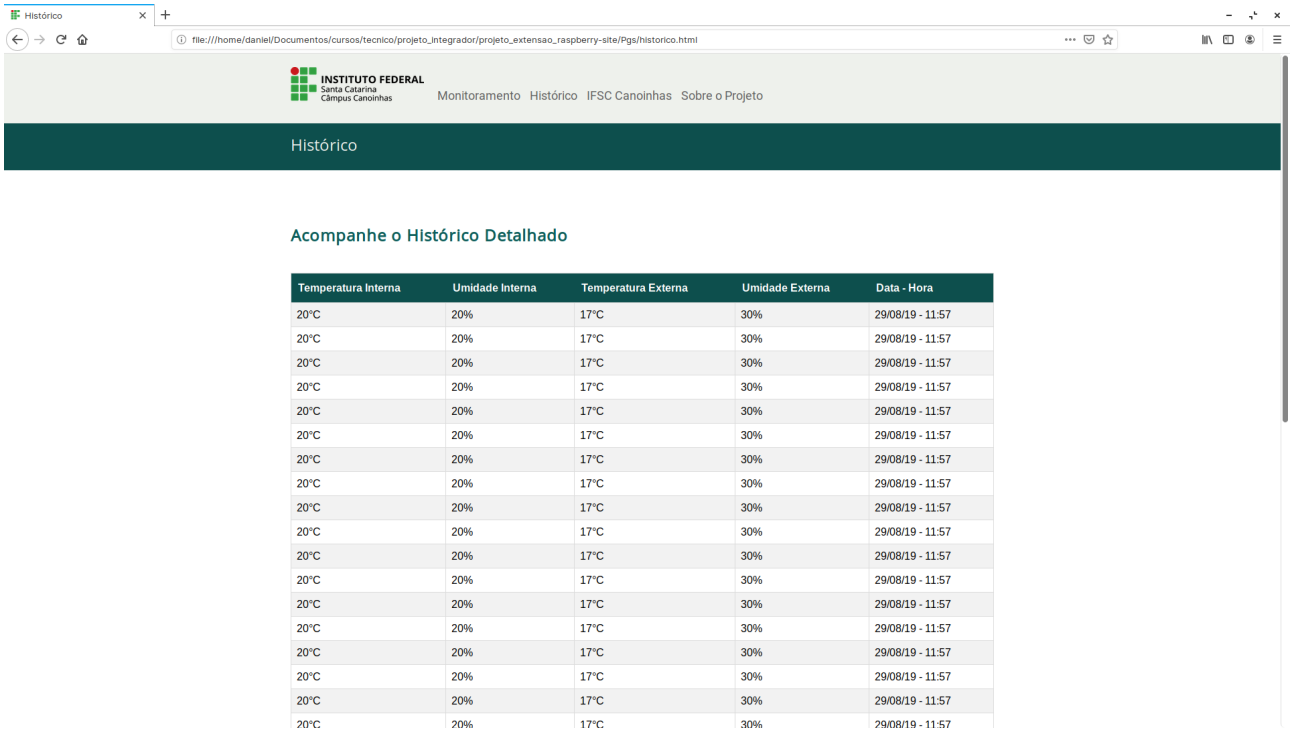


Figura 13: Página de histórico do site

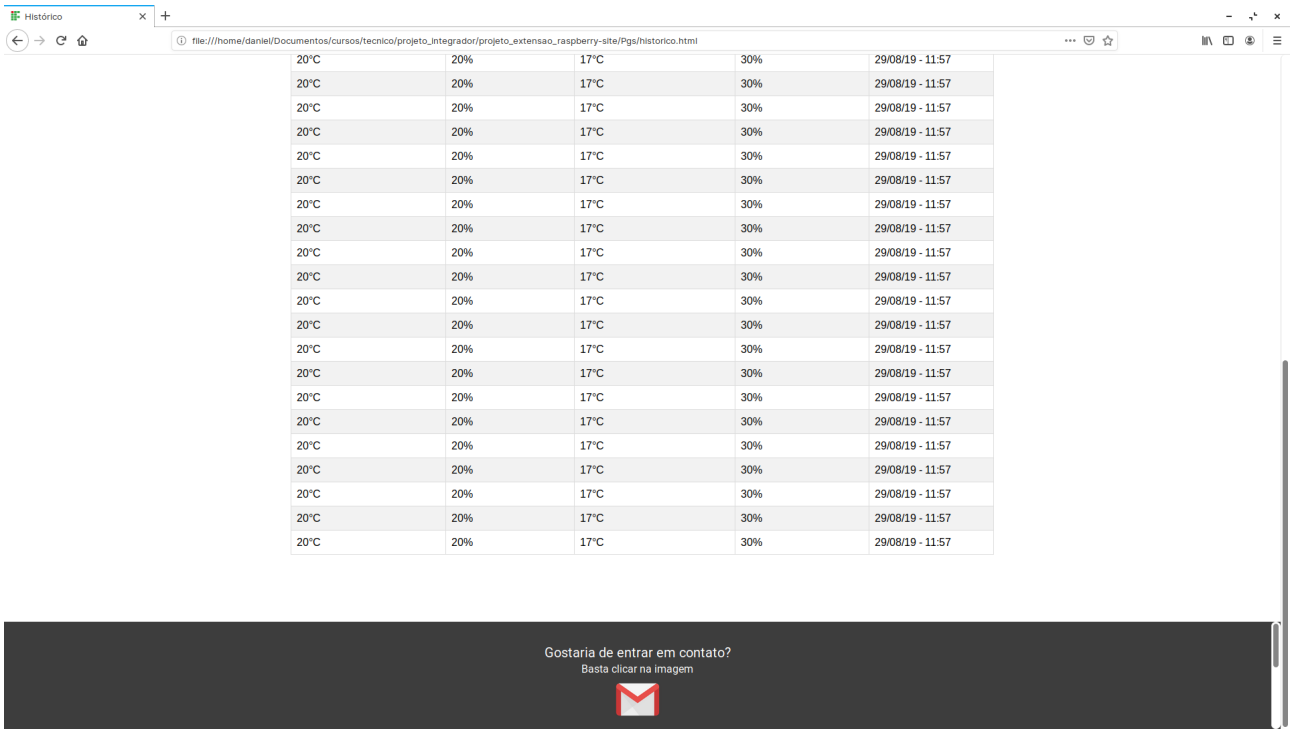


Figura 14: Página de histórico 2

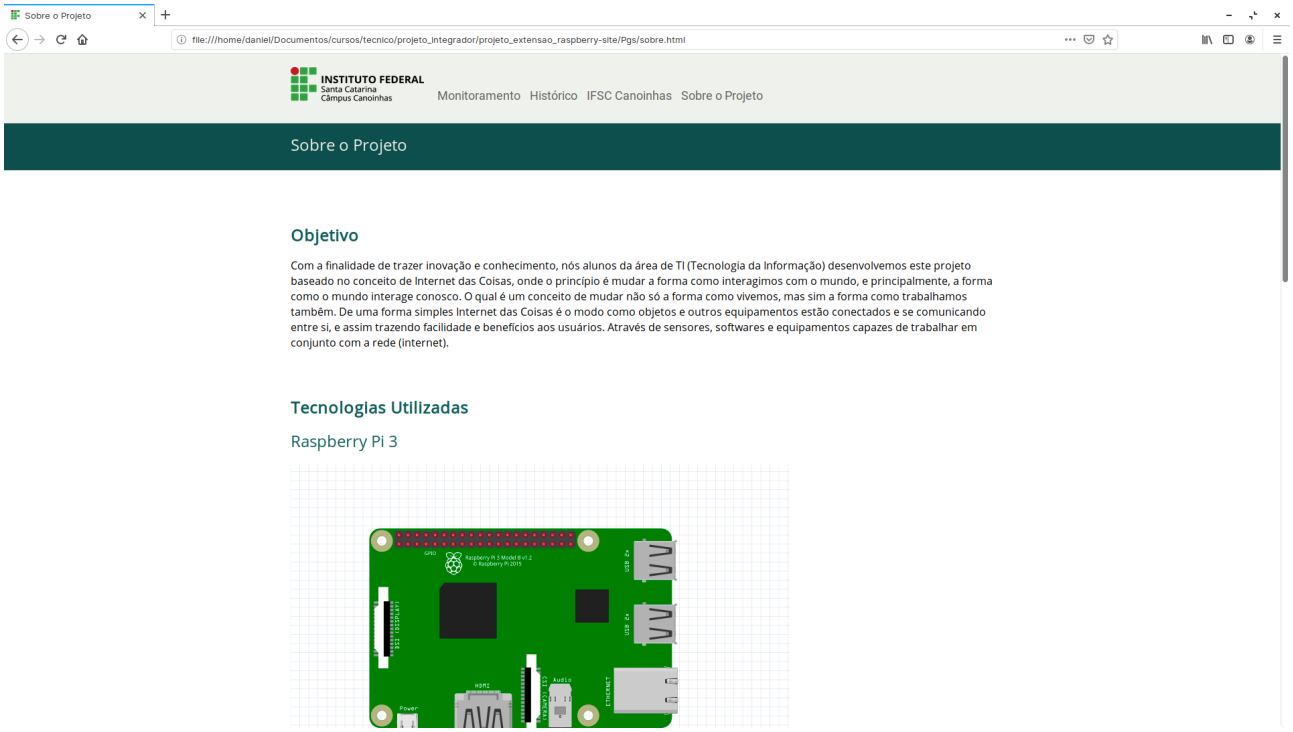


Figura 15: Página apresentado o projeto

4 ANÁLISE E DISCUSSÃO DOS RESULTADOS

Com o Raspberry fazendo a leitura dos dados a cada minuto, durante um dia temos 1440 leituras, todas estas leituras são salvas no banco de dados, cada vez que é enviada uma nova instância para a tabela, são usados 12 bytes de memória 4 campos do tipo (*TINYINT* - 1 byte) e um campo do tipo (*DATETIME* - 8 bytes), durante um dia se nenhuma leitura falhar são usados 17280 bytes de memória ou seja 17,28 Kilobytes, em um ano se o sistema funcionasse sem falhas seriam usados em torno de 6,3072 Megabytes.

Durante o período que deixamos fazendo os testes conseguimos em torno de 11000 registros no banco de dados.

Query 1 × registros × teste ×

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

#	temp_interna	temp_externa	umid_interna	umid_externa	dia
11241	28	23	95	127	2019-10-14 09:58:20
11242	28	23	95	127	2019-10-14 09:59:05
11243	28	22	95	127	2019-10-14 10:00:05
11244	28	23	95	127	2019-10-14 10:01:08
11245	28	23	95	127	2019-10-14 10:02:05
11246	28	23	95	127	2019-10-14 10:03:20
11247	29	23	95	127	2019-10-14 10:04:13
11248	29	23	95	127	2019-10-14 10:05:05
11249	29	23	95	127	2019-10-14 10:06:05
11250	29	23	95	127	2019-10-14 10:07:05
11251	30	24	95	127	2019-10-14 10:08:13
11252	30	24	95	127	2019-10-14 10:09:05
11253	31	25	95	127	2019-10-14 10:11:27
11254	31	25	95	127	2019-10-14 10:12:05
11255	31	25	95	127	2019-10-14 10:13:05
11256	31	49	95	19	2019-10-14 10:15:25
11257	31	24	95	127	2019-10-14 10:16:07
11258	30	24	95	127	2019-10-14 10:17:05
11259	29	23	95	127	2019-10-14 10:19:05
11260	30	23	95	127	2019-10-14 10:21:05
11261	30	23	95	127	2019-10-14 10:22:40
11262	30	22	95	127	2019-10-14 10:23:15
11263	30	22	95	127	2019-10-14 10:24:05
11264	30	22	95	127	2019-10-14 10:25:08
11265	30	22	95	127	2019-10-14 10:26:10
11266	30	22	95	127	2019-10-14 10:27:08
11267	31	23	95	127	2019-10-14 10:28:18
11268	31	23	95	127	2019-10-14 10:29:05
11269	31	23	95	127	2019-10-14 10:32:25
11270	31	23	95	127	2019-10-14 10:33:05

teste 1 ×

uery Completed

Figura 16: Registros no Banco de Dados

4.1 Funcionamento esperado dos sensores

Conforme o esperado conseguimos realizar o processo de ler os dados através dos sensores e enviá-los para o banco de dados, funcionou perfeitamente por determinado período de temo, mas porem o sensor externo acabou apresentando defeito na leitura de umidade, como podemos ver na figura 12 suas leituras sempre resultava em 127% de umidade o que ultrapassa o limite de 100% que seria o máximo, então não podemos considerar os dados obtidos de umidade externa para futuras comparações, os únicos dados confiáveis são os de temperatura interna e externa. Ao analisar superficialmente estes dados percebemos que a diferença entre a temperatura externa e interna é de cerca de 8 há 20 graus conforme o momento, então podemos perceber que durante todo o tempo a estufa mantêm uma temperatura superior a externa.

4.2 Problemas de aquecimento no Raspberry

Por conta da carcaça da antena utilizada ser fechada para garantir que não entre água, isso acabou gerando um superaquecimento no Raspberry em certos momentos de calor ao superaquecer o Raspberry desliga, e somente quando a temperatura de seu processador abaixa ele liga novamente, percebemos durante o período de testes que quando ocorria seu desligamento demorava em torno de 20 minutos para a temperatura abaixar e ele ligar novamente, o período em que ele funcionava melhor era durante a noite e madrugada, onde a temperatura era menor.

5 CONCLUSÃO

Apesar de ter ocorrido o problema com o sensor interno e o problema de super aquecimento do Raspberry, foram atingidos todos os objetivos conforme o esperado, notamos que é possível aplicar a TI em locais que fogem da nossa área de atuação, e que talvez nem imaginávamos no passado que poderíamos tentar prover uma solução para tal problema. Muitos conhecimentos foram adquiridos durante o processo, através do que foi aprendido durante o curso como lógica de programação, banco de dados, redes, eletrônica e configuração de serviços e servidores, com um pouco mais de pesquisa foi possível unir todos esses conhecimentos e produzir algo que tentasse suprir uma necessidade real. Partimos do inicio onde havia somente uma ideia, descobrimos uma necessidade, realizamos pesquisas e estudos para tentar prover uma solução eficaz, pela primeira vez lidamos com o Raspberry Pi, sensores e colocamos um banco de dados em produção, criamos um sistema capaz de realizar uma tarefa simples, o que pode parecer fácil agora levou muito tempo de estudo pesquisa e dedicação para ser elaborado, mesmo não sendo um sucesso obtivemos um excelente conhecimento para que na próxima vez possamos gastar menos tempo no processo já feito e mais tempo em novos problemas e desafios, assim tentando cada vez mais gerar um produto melhor.

REFERÊNCIAS

WIKIPÉDIA. Desenvolvido pela Wikipédia Foundation. Apresenta conteúdo enciclopédico. Disponível em: Acesso em: https://pt.wikipedia.org/wiki/Internet_das_coisas. 05 dez 2019.

WIKIPÉDIA. Desenvolvido pela Wikipédia Foundation. Apresenta conteúdo enciclopédico. Disponível em: https://pt.wikipedia.org/wiki/Agricultura_de_precis%C3%A3o. Acesso em: 05 dez 2019.

HCODE. Video aula sobre criação de usuário remoto. **Youtube**, 01 mai. 2019. Disponível em https://youtu.be/-MiPkEfRWJE . Acesso em: 05 out. 2019.

DESCONHECIDO, Flávio. Video aula sobre como iniciar na programação em Python no Raspberry Pi. **Youtube**, 22 fev. 2018. Disponível em https://youtu.be/0Z_-KNRLTG8. Acesso em: 04 set. 2019.

AFONSO, Miguel. Video aula sobre sensor DHT11. **Youtube**, 05 mai. 2018. Disponível em https://youtu.be/nTomhTSao4g. Acesso em: 03 set. 2019.

GUANABARA, Gustavo. Video aula 01 sobre Python. **Youtube**, 03 abr. 2017. Disponível em https://youtu.be/S9uPNppGsGo. Acesso em: 20 ago. 2019.