

**UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
CURSO DE ENGENHARIA DE CONTROLE E AUTOMAÇÃO
TRABALHO DE CONCLUSÃO DE CURSO**

Augusto Kucmanski

**MONITORAMENTO E ARMAZENAMENTO DA VARIAÇÃO DE
TEMPERATURA E UMIDADE EM UMA RESIDÊNCIA EFICIENTE**

Santa Maria, RS, Brasil

2018

Augusto Kucmanski

**MONITORAMENTO E ARMAZENAMENTO DA VARIAÇÃO DE TEMPERATURA E
UMIDADE EM UMA RESIDÊNCIA EFICIENTE**

Relatório apresentado ao Curso de Graduação em Engenharia de Controle e Automação da Universidade Federal de Santa Maria (ÚFSM, RS), como requisito para obtenção da nota final da disciplina de **Trabalho de Conclusão de Curso**.

Orientador: Prof. Dr. Frederico Menine Schaf

Coorientador: Prof. Dr. Marcos Alberto Oss Vaghetti

Santa Maria, RS

2018

RESUMO

MONITORAMENTO E ARMAZENAMENTO DA VARIAÇÃO DE TEMPERATURA E UMIDADE EM UMA RESIDÊNCIA EFICIENTE

AUTOR: Augusto Kucmanski

ORIENTADOR: Frederico Menine Schaf

COORIENTADOR: Marcos Alberto Oss Vaghetti

Em um país tropical como o Brasil, os estados, predominantemente, possuem um clima quente e úmido, o que vem ao encontro do problema considerado para a criação deste projeto: a ideia do sensoriamento da temperatura ambiente, durante 24 horas, em cada cômodo de uma casa eficiente, situada na UFSM (Universidade Federal de Santa Maria). Através dos dados obtidos, se conduziu de um estudo baseado nas temperaturas registradas, o que possibilita demonstrar qual a melhor escolha para a utilização dos cômodos de uma casa. Neste trabalho será desenvolvido um protótipo utilizando Arduino e seus componentes. O protótipo será constituído de duas partes, sendo a primeira com a utilização de IoT através da conexão Wi-Fi com um ESP8266-01 e a segunda com o emprego de um datalogger utilizando um relógio de alta precisão (RTC DS3231), um leitor de cartão micro SD, quatro sensores de temperatura e umidade (DHT22) e dois sensores de temperatura (DS18B20). Os dados coletados pelo protótipo serão armazenados tanto na forma online quanto na forma off-line. Para auxiliar na interpretação dos dados online serão utilizadas tabelas e gráficos que são atualizados em tempo real, para parte off-line, os dados serão salvos em blocos de notas que através de macros para o software Excel, podem ser interpretados mais facilmente. O protótipo teve um bom desempenho nos testes, sendo é necessária algumas melhorias para que no futuro possa ser utilizado de maneira mais profissional, como a soldagem de uma placa para não utilizar uma protoboard.

Palavras chaves: Datalogger, Temperatura, Iot, Automação Residencial, Arduino.

ABSTRACT

AUTHOR: Augusto Kucmanski

ADVISOR: Frederico Menine Schaf

CO-ADVISOR: Marcos Alberto Oss Vaghetti

In a tropical country like Brazil, most of the states' climate are predominantly hot and humid, such characteristic meets the problem considered for the proposal of this project: the idea of measuring the ambience temperature in each room of an efficient house located in UFSM for 24 hours. This aims to determine which rooms are the coolest by using researches such as the one of the house facing toward north. The data for this project was obtained via measuring hardware developed mostly by Arduino and its components: (1) first by the use of IoT via Wi-Fi connection with an ESP8266-01, and then (2) with a datalogger system made out of a high precision clock (RTC DS3231), micro SD card reader, four temperature and humidity sensors (DHT22) and two temperature sensors (DS18B20). Data was stored both in off-line backup and in the cloud. Charts and graphics were made to help interpreting the online data which is updated in real time while the off-line backup data is saved in NotePad files (.txt) that can be interpreted more accurately via Excel software macros. Test results were good, however some enhancements like soldering a board instead of using a protoboard must be performed in order to allow this experiment to be more proficiently.

Palavras chaves: Datalogger, Temperature, Low Cost, Residencial Automotion.

LISTA DE FIGURAS

Figura 1- Transformação dos sinais	16
Figura 2 - Conversão dos Sinais	16
Figura 3 - Casa virada para o norte (VAGHETTI et al., 2013).....	23
Figura 4 - ESP8266-01 (FILIPEFLOP, 2016).....	24
Figura 5 - Comandos AT enviados via Arduino	25
Figura 6 - DHT22 (FILIPEFLOP).....	26
Figura 7 - Sensor DS18B20 (ARDUINOBR,2015)	27
Figura 8 - Módulo cartão micro SD (FILIPEFLOP,2016)	28
Figura 9 - RTC DS3231 (FILIPEFLOP, 2016).....	29
Figura 10 - Cabo Multivias.....	29
Figura 11 - Arquitetura global do sistema	30
Figura 12 - Fluxograma	32
Figura 13 - Esquemático da montagem	33
Figura 14 - Cabo Extensor Multivias	33
Figura 15 - Planta baixa.....	34
Figura 16 – Criação do canal	35
Figura 17 - Informações do canal	35
Figura 18 - Chave de escrita	36
Figura 19 - Código Arduino	36
Figura 20 - Código Arduino	37
Figura 21 - Código Arduino	37
Figura 22 - Gráficos no website	38
Figura 23 - Exemplo de Planilha	38
Figura 24 - Configuração da planilha	39
Figura 25 - Script da planilha	39
Figura 26 - URL do documento.....	40
Figura 27 - Configuração script.....	40
Figura 28 - Configuração do script.....	41
Figura 29 - Declaração em modo público	41
Figura 30 - URL gerada.....	42
Figura 31 - Modificação do parâmetro através da URL.....	42
Figura 32 - Demonstração do resultado.....	42

Figura 33 - Serviço de requisição	42
Figura 34 - Configuração do serviço de requisição	43
Figura 35 - Criando dispositivo	43
Figura 36 - Adicionando tarefa ao dispositivo	43
Figura 37 - Configuração da tarefa.....	44
Figura 38 – DeviceID	44
Figura 39 - Resultado final	44
Figura 40 - Código Arduino	45
Figura 41 - Código Arduino	45
Figura 42 - Resultado da planilha Google	46
Figura 43 - Extração do WordPress no WAMP	46
Figura 44 – Login	47
Figura 45 - Configuração da criação da base de dados	47
Figura 46 - Configurações do banco de dados	48
Figura 47 - Configuração do administrador	48
Figura 48 - Interface do WordPress.....	49
Figura 49 - Configuração da página inicial	49
Figura 50 - Resultado do website	50
Figura 51 - Busca dos domínios	51
Figura 52 - Domínio utilizado	51
Figura 53 - Modificação dos servidores	51
Figura 54 - Plugin All-in-One	52
Figura 55 - Download do arquivo gerado.....	52
Figura 56 - Criação da conta no site InfinityFree.....	53
Figura 57 - Instalação do WordPress no novo site	54
Figura 58 - Acesso do website via domínio	54
Figura 59 - Sincronia do RTC	55
Figura 60 - Código do Arduino para criação do documento	55
Figura 61 - Configuração da data e hora	56
Figura 62 - Configuração temperatura e umidade	56
Figura 63 - Bloco de notas criado.....	56
Figura 64 - Conteúdo do bloco de notas.....	57
Figura 65 - Ambiente VBA	57
Figura 66 - Botões	58

Figura 67 - Execução da primeira macro.....	58
Figura 68 - Resultado da importação.....	59
Figura 69 - Resultados nas planilhas Google no início	59
Figura 70 - Resultados nas planilhas Google no final	60
Figura 71 - Gráfico da temperatura da cozinha	61
Figura 72 - Gráfico da umidade da cozinha	61
Figura 73 - Gráfico da temperatura da sala	62
Figura 74 - Gráfico da umidade da sala.....	62
Figura 75 - Gráfico da temperatura do quarto dos fundos.....	63
Figura 76 - Gráfico da umidade do quarto dos fundos	64
Figura 77 - Gráfico da temperatura do quarto da frente	65
Figura 78 - Gráfico da umidade do quarto da frente	65
Figura 79 - Gráfico da temperatura externa.....	66
Figura 80 - Temperatura do banheiro	67
Figura 81 - Sistema externo de calefação	69
Figura 82 - Sistema interno de calefação.....	69

LISTA DE TABELAS

Tabela 1 - Conexão com o Arduino.....	31
Tabela 2 - Comprimento dos cabos	34
Tabela 3 - Máximos e mínimos da cozinha.....	60
Tabela 4 - Máximos e mínimos da sala	62
Tabela 5 - Máximos e mínimos do quarto dos fundos	63
Tabela 6 - Máximos e mínimos do quarto da frente.....	64
Tabela 7 - Máximos e mínimos da temperatura externa	66
Tabela 8 - Temperaturas máximas e mínimas de 2015	67
Tabela 9 - Tabela temperaturas de 2018.....	68

SUMÁRIO

1.	INTRODUÇÃO.....	11
1.1	DESCRIÇÃO DO PROBLEMA	12
1.2	OBJETIVOS	12
2.	REVISÃO DE LITERATURA	13
2.1	IOT.....	13
2.2	BIG DATA	14
2.3	INDÚSTRIA 4.0.....	14
2.4	AQUISIÇÃO DE DADOS	15
2.5	DATALOGGER	17
2.6	ARDUINO.....	18
2.7	LINGUAGEM E AMBIENTE DE PROGRAMAÇÃO.....	18
2.8	THINGSPEAK	19
2.9	PUSHINGBOX.....	19
2.10	PROTOCOLO DE COMUNICAÇÃO TCP.....	20
2.11	LINGUAGEM HTML.....	20
2.12	LINGUAGEM JAVA.....	21
2.13	MICROSOFT EXCEL.....	21
2.14	WORDPRESS	21
2.14.1	Plug-ins Utilizados.....	22
2.15	WAMP SERVER.....	22
2.16	WEBSITE INFINITY FREE	22
2.17	ESTUDO DA CASA VIRADA PARA O NORTE	23
3.	MATERIAIS E MÉTODOS	24
3.1	ESP8266-01	24
3.1.1	Comandos	24
3.2	SENSOR DE TEMPERATURA E UMIDADE DHT22.....	26
2.3	SENSOR DE TEMPERATURA DS18B20.....	26
2.4	MODULO DE CARTÃO MICRO-SD.....	27
2.5	MÓDULO RTC DS3231	28
2.6	CABO MULTIVIAS	29
4.	RESULTADOS	30
4.1	ARQUITETURA GLOBAL DO SISTEMA.....	30
4.2	CONFIGURAÇÕES DO SISTEMA AO THINGSPEAK	35
4.2	CONEXÃO COM A PLANILHA GOOGLE.....	38
4.3	CRIAÇÃO DO WEBSITE	46

		10
4.3.1	Colocando o website online	50
4.4	PROGRAMAÇÃO DO DATALOGGER	55
4.6	RESULTADOS NA CASA EFICIENTE	59
4.6.1	Resultados da cozinha	60
4.6.2	Resultados da sala	61
4.6.3	Resultado do quarto dos fundos	63
4.6.4	Resultado do quarto da frente.....	64
4.6.5	Resultados da temperatura externa.....	66
4.6.6	Resultados do banheiro	66
4.7	COMPARAÇÃO COM RESULTADOS ANTERIORES	67
4.8	MATERIAIS E CUSTOS	70
5.	CONCLUSÃO	71
	REFERÊNCIAS	76

1. INTRODUÇÃO

A Internet sem dúvida foi um dos maiores avanços tecnológicos para o mundo moderno, revolucionando as nossas vidas. Segundo uma pesquisa do Instituto Brasileiro de Geografia e Estatística (IBGE), quando voltamos para 2005, o número de residências que possuía conexão com a Internet era de apenas 20,9%, nos atuais dias, 13 anos depois, esse número triplicou, subindo para aproximadamente 64%.

Não foi apenas nas residências brasileiras que esse número aumentou, nas indústrias, com o auxílio da Internet, houve uma grande evolução da maneira como podemos registrar e visualizar dados coletados de sensores, fazendo com que a Internet seja muito atrativa. Uma pesquisa realizada pela Associação Brasileira de Internet Industrial (ABII), revelou que em 2017 a Internet industrial no Brasil encontra-se em período de transição e implementação, pois o país encontra-se na fase de automação, através da eletrônica, robótica e programação, características da Terceira Revolução Industrial sendo que o cenário para os anos futuros é apenas de crescimento.

Quando mencionamos sobre a situação da Internet na indústria brasileira, não podemos deixar de citar a Internet das Coisas, conhecida pela sigla *IoT*, que deriva do inglês *Internet of Things*. Sua definição é basicamente a comunicação de objetos via Internet, que permite não só o compartilhamento de informações como também o controle, seja de uma linha de produção em uma fábrica, ou um sistema de monitoramento de uma residência. É graças a ela que podemos supervisionar e armazenar dados de um processo em tempo real, também pode-se monitorar problemas e execuções.

Outra maneira simples de armazenar dados é um *datalogger*, que possui uma unidade de armazenamento responsável para registrar o que se deseja medir, muito eficiente em casos de difícil conexão de aparelhos com a Internet.

Muito pode-se monitorar em uma simples residência, seja desde níveis de caixas de água, luminosidade de uma sala, temperatura e umidade de um cômodo, ou até mesmo câmeras para segurança. Uma temperatura amena é algo primordial quando o assunto é conforto dentro de uma residência, tendo estudos que confirmem o melhor planejamento para a construção de uma casa, fazendo com que as temperaturas sejam mais adequadas durante o dia.

1.1 DESCRIÇÃO DO PROBLEMA

Com o advento das tecnologias tanto na parte industrial como na parte residencial, houve uma necessidade de um sensoriamento preciso para monitorar os dados das mais diversas aplicações. Mas de nada serve um perfeito sensoriamento se não podemos analisar a informação com facilidade e eficiência.

Um aparelho para adquirir dados é um dispositivo tecnológico que registra informações transmitidas por outros dispositivos, como sensores e atuadores, armazenando-os para *download*. Um exemplo disso, é um *datalogger* que opera sem que seja necessário a utilização de um computador, o que lhes permite autonomia e melhor utilização do espaço disponível.

Mas, quando deseja-se realizar medições em tempo real, utilizar a *IoT* é uma das maneiras mais eficientes para desempenhar essa função, superando a utilização de um *datalogger*, que infelizmente não faz o sensoriamento em tempo real, sendo necessário exercer a remoção e a leitura do cartão de memória em um computador para verificar se há ou não alguma divergência com relação as variáveis medidas.

1.2 OBJETIVOS

Aplicar os conhecimentos adquiridos ao longo da graduação do curso de Engenharia de Controle e Automação para desenvolver um sistema de monitoramento preciso utilizando *IoT*, com auxílio de ferramentas gratuitas como as planilhas do *Google* e sites como *ThingSpeak*, para gerar gráficos e estudos com base nas temperaturas registradas. Unir todo o conteúdo adquirido em um *site* desenvolvido através da plataforma *WordPress*, para se ter melhor comodidade e eficiência tanto no armazenamento quanto na análise dos dados obtidos

Também se tem o objetivo de guardar os dados, mas em um cartão de memória que vai agir como um *datalogger*, com o intuito de um armazenamento mais completo e seguro para caso haja quedas de conexão durante as medições.

2. REVISÃO DE LITERATURA

Neste capítulo será apresentada uma fundamentação teórica do estudo, como mais informações sobre a *IoT*, descrevendo como é feita as conexões via *Wi-Fi* e o que foi utilizado.

2.1 IOT

A ideia de dispositivos conectados surgiu nos anos 70, nesta época, costumava ser denominado de “Internet embutida” ou “computação generalizada”. O atual termo “Internet das Coisas” foi dado por Kevin Ashton, em 1999, quando era empregado na *Procter & Gamble*. Mas, foi somente em 2010 que o conceito começou a se tornar popular mundialmente, quando vazou uma informação de que o serviço *StreetView*, da *Google*, não só criou fotos de 360 graus, mas também armazenou uma enxurrada de dados das redes *Wi-Fi* das pessoas (MAGNUS 2017).

A Internet das Coisas é a nova revolução no setor industrial e está capacitando a engenharia industrial com a ajuda de sensores, *softwares* e análise de dados para criar máquinas muito eficientes.

A filosofia de condução por trás da *IoT* é que a transformação digital, com as máquinas inteligentes, é mais precisa e consistente do que os humanos na comunicação por meio de dados, e esses dados podem ajudar as empresas a detectarem ineficiências e problemas mais cedo.

Os exemplos de Internet de Coisas se estendem desde casas conectadas inteligentes até cuidados com o bem-estar, pois hoje em dia, a *IoT* está se tornando cada vez mais parte de todos os aspectos de nossas vidas, porque não só as aplicações da Internet das Coisas melhoram a nossa comodidade, mas também nos dão mais controle para simplificar as funções da vida profissional e as tarefas pessoais.

Como por exemplo, a *Nike* e a *Apple* desenvolveram um sensor, para ser colocado na parte inferior da palmilha de um tênis. Usando o *smartphone* ou *Ipod*, o corredor pode definir a distância a qual se deseja correr, quantas calorias é pretendido perder, o seu percurso e até mesmo uma lista de músicas para ouvir. Ao terminada a atividade, todas as informações são

enviadas automaticamente para um *site*, onde é possível acompanhar o histórico de corridas, a evolução e até dividir resultados com outros corredores conectados (ENDEAVOR).

2.2 BIG DATA

O termo *Big Data* significa a análise e a compreensão de uma grandes quantidade de dados dos mais variados tipos. Embora o termo seja consideravelmente novo, a prática de coletar e armazenar grandes proporções de dados para estudos freqüentes é muito antigo. O termo começou a ganhar força no começo dos anos 2000, quando o analista Doug Laney articulou a definição atualmente conhecida de *big data* em três características (SAS 2015):

- “Volume: Organizações coletam dados de fontes variadas, incluindo transações financeiras, mídias sociais e informações de sensores ou dados transmitidos de máquina para máquina. No passado, armazená-los teria sido um problema — mas novas tecnologias aliviam esse fardo.”
- “Velocidade: Os dados são transmitidos numa velocidade sem precedentes e devem ser tratados em tempo hábil. Identificação por radiofrequência, sensores e medições inteligentes estão impulsionando a necessidade de lidar com torrentes de dados praticamente em tempo real.”
- “Variedade: Dados são gerados em inúmeros formatos — desde estruturados (numéricos, em *databases* tradicionais) a não-estruturados (documentos de texto, e-mail, vídeo, áudio, cotações da bolsa e transações financeiras).”

Vale salientar que a importância do *Big Data* não se é em torno da quantidade de dados obtidos, mas sim do que pode ser feito com eles. Podendo obter informações de várias fontes e analisá-los para encontrar respostas que permitem diminuir recursos, otimizar tempo, desenvolver novos produtos e tomar decisões mais inteligentes (SAS,2015).

2.3 INDÚSTRIA 4.0

O termo indústria 4.0 teve sua origem a partir de um projeto do governo alemão voltadas para o desenvolvimento da tecnologia, ele foi utilizado primeiramente numa feira denominada *Hannover* no ano de 2011. Tendo como o princípio básico a conexão das máquinas, sistemas e ativos, que as companhias e empresas poderão usar para criar redes inteligentes, assim podendo controlar as máquinas da produção de forma autônoma. Ou seja,

“as fábricas inteligentes terão a capacidade e autonomia para agendar manutenções, prever falhas nos processos e se adaptar aos requisitos e mudanças não planejadas na produção” (SILVEIRA,2017).

Um dos maiores impactos causados pela indústria 4.0 será uma mudança que afetará o mercado como um todo, que consiste na criação de novos modelos de negócios em um mercado cada vez mais exigente, muitas empresas já procuram integrar ao produto necessidades e preferências específicas de cada cliente. A customização prévia do produto por parte dos consumidores tende a ser uma variável a mais no processo de manufatura, mas as fábricas inteligentes serão capazes de levar a personalização de cada cliente em consideração, se adaptando às preferências.

Com base no mesmo assunto, outra área que será abalada pela quarta revolução industrial será a área de pesquisa e desenvolvimento nos campos de confiabilidade da produção, segurança em tecnologia e informação (T.I.) e interação máquina-máquina. Esta tecnologia deverá ser desenvolvida continuamente para que se torne possível a adaptação de empresas para este novo padrão de indústria que está nascendo. Os trabalhadores também precisarão de uma adaptação, pois com fábricas ainda mais inteligentes, novas oportunidades surgirão enquanto algumas deixarão de existir. “Os trabalhos manuais e repetitivos já vem sendo substituídos por mão de obra automatizada, e com indústria 4.0 isso tende a continuar. Por outro lado, as demandas em pesquisa e desenvolvimento oferecerão oportunidades para profissionais tecnicamente capacitados, com formação multidisciplinar para compreender e trabalhar com a variedade de tecnologia que compõe uma fábrica inteligente” (SILVEIRA,2017).

2.4 AQUISIÇÃO DE DADOS

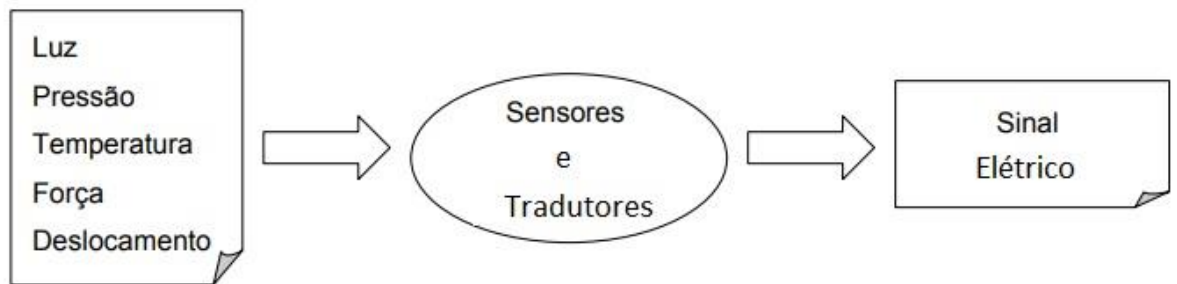
A aquisição de dados é algo primordial quando se deseja realizar qualquer tipo de estudo, sobre algum assunto em geral. Pois é a partir dela que pode-se começar a realizar um projeto, e para que este projeto seja confiável, é necessário uma boa aquisição de dados, não tendo nenhum tipo de perdas ou interferência.

A obtenção dos dados está presente em quase todas as atividades diárias, estamos cercados por toda parte, por diversos tipos de sistemas que recolhem informações e auxiliam no processo da tomada de decisão. Adquirir dados pode, de maneira simples, ser definida

como a medição de informações do mundo real, e a grande parte dos eventos do mundo real possui natureza analógica, isto é, a medição pode conduzir a uma gama de valores contínuos.

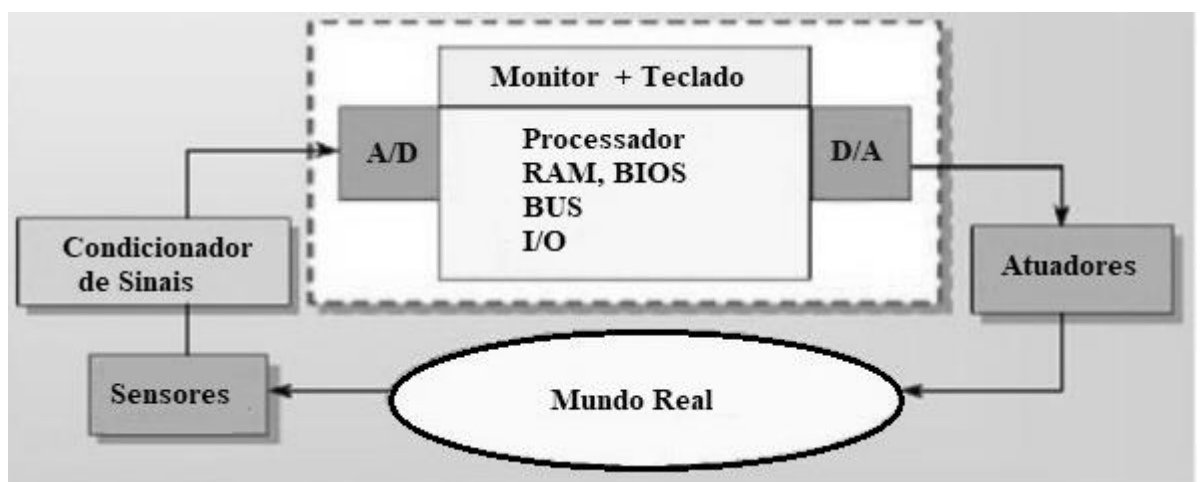
Todas as grandezas possuem uma determinada energia, deste modo, torna-se necessário para sua mensuração a utilização de dispositivos capazes de receber esta energia, relativa a uma determinada quantidade física da grandeza desejada para então convertê-la numa forma de energia manipulável pelos circuitos eletrônicos. Estes dispositivos são os sensores e transdutores. Os sensores e transdutores recebem as quantidades físicas de grandezas analógicas e convertem-nas em quantidades elétricas, tais como tensão, corrente ou impedância, como pode-se observar no exemplo da Figura 1 (BAPTISTA, 2013).

Figura 1- Transformação dos sinais



Para tanto, esses sistemas devem apresentar uma arquitetura onde os elementos conseguem realizar a comunicação mútua, interagindo entre si. Significando que um sinal gerado por um sensor ou transdutor pode ser analisado pelo condicionador de sinais, que tem por função entregar um diferente sinal, relacionado com o primeiro e que pode ser tratado pelo conversor analógico-digital e assim sucessivamente. Um exemplo claro disto, está na Figura 2, que representa o caminho percorrido para a conversão das variáveis do mundo real (MARTINS, 2013).

Figura 2 - Conversão dos Sinais



2.5 DATALOGGER

O *datalogger* é um equipamento que ajuda a coleta e armazenamento de informações de outros instrumentos, através de um sistema de contagem de pulsos eletrônicos. Possui uma unidade de memória que permite reter essas informações coletadas e pode se comunicar com outros dispositivos através de *softwares* adequados para este fim (NASCIMENTO,2010).

Possui uma série de vantagens, como as listadas abaixo:

- Aumentam as possibilidades de monitoramentos ambientais com objetivos específicos;
- O processo de coleta e armazenagem dos dados são mais rápidos e muitas vezes mais precisos do que os manuais;
- Obtenção de dados com maior frequência, o que melhora a qualidade dos dados obtidos;
- Possibilidade de transferência dos dados para um computador;
- Podem funcionar de maneira autônoma sem necessidade de alimentação externa.

O *hardware* do *datalogger* é composto por seis módulos, segundo Nascimento (2010):

1. Processador

É responsável pela execução dos cálculos e em coordenar todas as tarefas.

2. Micro controlador

É um componente que já tem incorporado vários blocos necessários de um sistema padrão de microprocessadores.

3. Modulo de comunicação serial

Interface que disponibiliza acesso à dados de leitura e escrita, via serial, com o microprocessador.

4. Memória

Acessa ou armazena as informações digitais mediante endereçamento em lugares denominados localidades de memória.

5. Conversor A/D (Analogico / Digital)

O conversor analógico-digital é um dispositivo eletrônico capaz de gerar uma representação digital a partir de uma grandeza analógica, normalmente um sinal representado por um nível de tensão ou intensidade de corrente elétrica.

6. Fonte de alimentação ou regulador de tensão

Fonte de alimentação para funcionamento do *datalogger*.

2.6 ARDUINO

Arduino é uma plataforma de computação física de código aberto baseado em *hardware* e *software* de fácil utilização, onde pode ser programada para processar as entradas e saídas entre o dispositivo e os componentes externos conectados a ele.

O *Arduino* foi o primeiro a lançar um *hardware* totalmente livre, o que atraiu muitas pessoas interessadas em desenvolver e contribuir com o mesmo. Ele é usado na construção de diversos projetos, como por exemplo os de automação e robótica, principalmente devido a incomplexidade de sua utilização, bem como também as comunidades ativas dispostas a compartilhar projetos, códigos e artigos na Internet que auxiliam muito na construção de um novo projeto (DILLY; MENDES. 2015).

Seu surgimento partiu da procura de um meio de baixo custo e a fim de facilitar a inserção de estudantes de *design* em tecnologia no *Interaction Design Institute* na cidade de Ivrea, Itália. O professor Massimo Banzi, em 2005, em conjunto com David Cuartielles, um pesquisador visitante da Universidade de Malmö, na Suécia, que também estava a procura de uma solução parecida, foram os que deram os primeiros passos ao criar algo que foi batizado de *Arduino*. “Banzi e Cuartielles então, decidiram desenvolver um micro controlador que poderia ser utilizado pelos seus estudantes de arte e *design* em seus projetos, com os principais fatores que fosse uma plataforma que qualquer pessoa pudesse utilizar e que fosse barato” (DILLY; MENDES, 2015).

2.7 LINGUAGEM E AMBIENTE DE PROGRAMAÇÃO

A linguagem de programação para o *Arduino* é baseada nas linguagens C/C++ e seus programas são chamados de *sketch*, ou seja, o arquivo que contém as linhas de códigos de instrução que são compiladas, enviadas e executadas em uma placa *Arduino*. A integração de comunicação do *Arduino* com qualquer outra aplicação de linguagem diferente se faz através dos componentes de comunicação que o *Arduino* dispõe (MCROBERTS, 2011).

“Um *sketch* típico consiste de duas partes ou rotinas: a primeira é a rotina de inicialização chamada *setup*, e a segunda é a rotina chamada *loop*, que geralmente contém o corpo principal do código” (EMBARCADOS, 2016).

O *Arduino* deve ser configurado previamente antes que possa trabalhar com ele, essas configurações são colocadas dentro de uma rotina de inicialização chamada *setup*. Coisas

típicas que devem ser feitas no *setup* incluem a inicialização dos pinos digitais e definição da taxa de transmissão para a comunicação serial (EVANS, NOBRE, HOCHENBAUM, 2013).

A função *setup* é sempre necessária, mesmo quando não for utilizada, pois caso ela não exista, ocorrerá um erro quando for verificado ou carregado o código. Já a segunda função pré-definida de um *sketch* do *Arduino* é a função chamada *loop*, que é executada logo após a função *setup* e a função é invocada continuamente enquanto o *Arduino* estiver alimentado de energia elétrica (EVANS, NOBRE, HOCHENBAUM, 2013).

O *open source Arduino Software (IDE)* é o responsável por compilar o código e enviar o mesmo para o micro controlador. O ambiente de desenvolvimento é compatível com os principais sistemas operacionais: *Windows*, *Mac OS X* e *Linux*. Em um desenvolvimento com *Arduino* normalmente são utilizados outros componentes eletrônicos, alguns simples como LED's e outros mais complexos como placas de rede, sensores, motores, entre outros componentes eletrônicos. No caso dos componentes mais complexos, é normal existir bibliotecas específicas para os componentes de forma a facilitar o desenvolvimento da mesma. (EVANS, NOBRE, HOCHENBAUM, 2013).

2.8 THINGSPEAK

“*ThingSpeak* é um serviço que oferece uma infraestrutura de *Web* e um protocolo de comunicação baseado em *HTTP* para envio e recebimento de dados gerados em *Arduino* ou qualquer outro dispositivo com recursos para comunicação em rede” (DR.BIT,2014). Todo dispositivo que se pode conectar com a Internet com serviço *HTTP* e capaz de realizar *GET* e *POST* pode se comunicar com a plataforma *ThingSpeak*. Com esse serviço é possível por exemplo usar o *Arduino* para gerar um *tweet*, monitorar variáveis, criar gráficos ou até mesmo criar uma rede social de dispositivos ou controlar a distância motores e outros dispositivos.

2.9 PUSHINGBOX

PushingBox é um sistema *online* que consegue enviar notificações baseadas em chamadas *Application Programming Interface (API)* ou em português, Interface de Programação de Aplicativos, que basicamente é a “matrix” por trás dos aplicativos. Em uma

requisição, pode-se enviar diversas notificações, desde um *Tweet*, até um e-mail, tudo isso em tempo real. Neste trabalho, será utilizado para mandar requisições em *HTTP*.

2.10 PROTOCOLO DE COMUNICAÇÃO TCP

O *TCP* (*Transmission Control Protocol* - Protocolo de Controle de Transmissão) é um dos principais protocolos da camada de transporte do modelo *TCP/IP*. Ele permite gerenciar os dados recebidos da (ou com destino à) camada inferior do modelo (ou seja, o protocolo *IP*). Quando os dados são fornecidos ao protocolo *IP*, este encapsula-os em datagramas *IP* ('datagrama' é um nome genérico para uma mensagem enviada sem conexão e sem confirmação, como os pacotes *IP*), fixando o campo do protocolo em 6 (para saber que o protocolo ascendente é o *TCP*). O *TCP* é um protocolo orientado para a conexão, isto é, ele permite que duas máquinas se comuniquem entre elas, além de controlar o estado da transmissão (CCM,2018).

As características do protocolo *TCP* são entregar ordenadamente os datagramas provenientes do protocolo *IP*, verificar a onda de dados para evitar uma saturação da rede, formatar os dados em segmentos de comprimento variável para 'entregá-los' ao protocolo *IP*, permitir o multiplex dos dados, ou seja, fazer circular, simultaneamente, as informações de fontes distintas na mesma linha e permitir o início e o fim de uma comunicação de maneira correta (CCM, 2018).

2.11 LINGUAGEM HTML

A linguagem *HTML* é uma linguagem popular mundialmente, é utilizada para o desenvolvimento de *Websites*, seu nome deriva do acrônimo em inglês *Hypertext Markup Language*, ou em português, Linguagem de Marcação de Hipertexto, ela é a "linguagem base da internet", o intuito de sua criação foi para o fácil entendimento, seja por humanos, seja por máquinas. (EIS, 2011).

Neste trabalho será utilizado o método de comunicação *GET* e *POST*. Quando utilizado o método *GET*, ocorre o envio de uma *string* anexada a uma *URL*, sendo relativamente mais simples, podendo transportar apenas uma *string*. Já quando utilizado o *POST*, a *string* é encapsulada junto ao corpo da requisição *HTTP*, ocorrendo uma perda de tempo no encapsulamento da mensagem, mas podendo transportar tanto textos quanto dados binários.

2.12 LINGUAGEM JAVA

Foi desenvolvida por uma equipe liderada por James Gosling na *Sun Microsystems* (atualmente de propriedade da *Oracle*) e lançada em 1995, o “Java é uma linguagem de programação orientada a objetos que atualmente faz parte do núcleo da Plataforma Java” (PEREIRA, 2009). Ela é utilizada para controlar o *HTML* e o *CSS*, manipulando os comportamentos na página.

2.13 MICROSOFT EXCEL

O *Microsoft Office Excel* é um editor de planilhas (Folhas de Cálculo) produzido pela *Microsoft*. Seus recursos incluem uma interface intuitiva e capacitadas ferramentas de cálculo e de construção de gráficos que tornou o *Excel* um dos programas mais populares de computador até hoje.

Para aproveitar toda a potência que o *Excel* possui, é indispensável a utilização de macros. “Macro poder ser definida como uma série de comandos (em sequência), que podem ser cliques, toques no teclado ou até mesmo pequenas linhas de códigos com funções mais avançadas. Essas sequencias são gravadas em um modulo *VBA* e são executados sempre que for necessário” (ÁVILA, 2015).

2.14 WORDPRESS

WordPress é um Sistema de Gestão de Conteúdo, também conhecido como *Content Management System* ou simplesmente *CMS*. Esse tipo de sistema fornece aos usuários “a capacidade de publicar, editar, modificar, organizar e deletar conteúdo por meio de uma interface central e única” (HOSTMEDIA, 2018).

O *WordPress* foi escrito em *PHP* com banco de dados *MySQL*, voltado sobretudo para a criação de *sites* e *blogs* via *Web*. Essa é uma das ferramentas mais famosas na criação de *blogs* disputando diretamente com o serviço do *Google*. É utilizado por quem queira um site mais profissional e com mais ferramentas a disposição (HOSTMEDIA, 2018).

2.14.1 Plug-ins Utilizados

Para melhor utilização da poderosa ferramenta que o *WordPress* é, não pode faltar a utilização dos mais diversos *plug-ins*. Neste trabalho foram utilizados diversos, que serão listados abaixo:

All-In-One WP Migration: Este *plug-in* tem a função de realizar todo um *backup* do *Website* criado *off-line*. Muito útil para quando se deseja exportar todos os arquivos com maior facilidade, seja por *backup* de segurança ou para colocá-lo *online* em outro servidor de hospedagem.

Widgets for ThingSpeak: Responsável por importar os gráficos feitos através do *site ThingSpeak*, auxiliando na facilidade para programação do *Website*.

SiteOrigin CSS: Facilitou as mudanças gráficas do *site*, como mudança de cores, fontes e estilo.

Title Remover: Como o próprio nome sugere, remove títulos desnecessários que estão programados automaticamente.

2.15 WAMP SERVER

WAMP é o termo usado para denominar os *softwares* que efetuam a instalação automática de vários *softwares* de forma que haja maior facilitação e agilidade para instalação. Em geral é usado *WAMP* para dizer que é um instalador de *Apache*, *Mysql* e *PHP* para *Windows*.

Para desenvolver aplicações utilizando *PHP* geralmente se utiliza três ferramentas básicas: o compilador da linguagem, um servidor e um banco de dados. Assim, para se preparar o ambiente de trabalho seriam necessários pelo menos três *downloads* e instalações: O *WampServer* é um pacote completo, nele obtemos: *Apache 2.2.22*, *MySQL 5.5.24*, *PHP 5.3.13* (DEV MEDIA, 2012).

2.16 WEBSITE INFINITY FREE

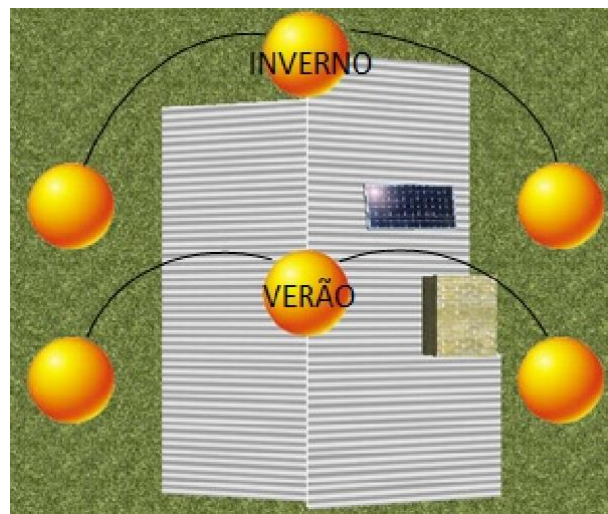
Utilizado para hospedagem grátis do *Website*, oferecendo banco de dados, tráfego ilimitado e subdomínios. Já possui algumas ferramentas para criação do *Website*, como o *WordPress*.

2.17 ESTUDO DA CASA VIRADA PARA O NORTE

Para que se tenha uma casa termicamente confortável, sendo ela mais quente no inverno e fresca no verão, alguns fatores devem ser observados: a insolação, as aberturas das janelas e a temperatura na edificação em diferentes horários do dia, entre outros.

As faces dos imóveis são bastante valorizadas na hora de sua compra. A face norte é aquela pela qual chega a maior quantidade de sol e, no inverno, os cômodos voltados para esse lado terão maior quantidade de luz solar durante mais tempo, o que poderá mantê-los aquecidos, mas, no verão, os espaços voltados para essa face receberão sol até o meio da tarde, o que pode reduzir o calor excessivo durante a noite.

Figura 3 - Casa virada para o norte (VAGHETTI et al., 2013).



3. MATERIAIS E MÉTODOS

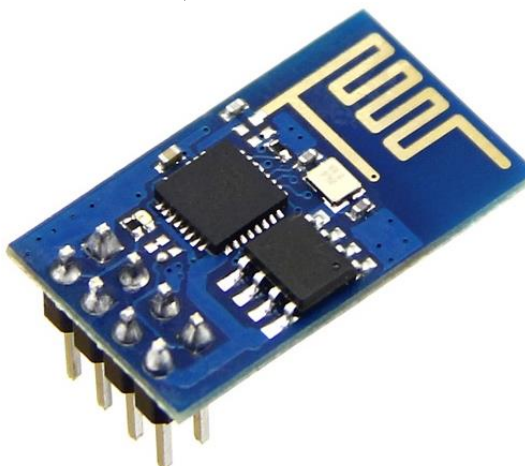
Neste capítulo são apresentados os materiais para a composição do trabalho, como placas e módulos.

3.1 ESP8266-01

O ESP8266 é um *chip*, de baixo custo e rápida disseminação. Responsável por realizar uma conexão *Wi-Fi* possibilitando a troca de dados entre diversos dispositivos via Internet (ou rede local) como sensores, atuadores e etc.

A versão ESP-01 é a versão mais comum do ESP8266, neste trabalho simplesmente referido como ESP, foi o primeiro módulo a ser disseminado, pois é o mais simples e com o menor custo em relação as demais versões. Sendo necessário ajuda de componentes externos para ser utilizado, como por exemplo o próprio *Arduino* para ser conectado.

Figura 4 - ESP8266-01 (FILIFELOP, 2016)



3.1.1 Comandos

No modo Comandos AT, o módulo possui instalado um programa chamado “*Firmware Comandos AT*”. Esse programa tem uma função pré-determinada que é receber comandos de operação via *universal asynchronous receiver/transmitter (UART)* (pinos *TX* e *RX*). Nesse

modo, o ESP-01 deverá trabalhar em conjunto com um micro controlador, sendo o micro controlador responsável por enviar os comandos AT para o ESP-01. O *Firmware* Comandos AT já está instalado de fábrica podendo estar em diferentes versões, mas podendo ser atualizado caso esteja com uma versão antiga.

Figura 5 - Comandos AT enviados via Arduino

```

AT

OK
AT+GMR
AT version:1.3.0.0(Jul 14 2016 18:54:01)
SDK version:2.0.0(5a875ba)
Farylink Technology Co., Ltd. v1.0.0.2
May 11 2017 22:23:58
OK
AT+CWMODE?
+CWMODE:1

OK
AT+CWLAP
+CWLAP:(4,"VIVO-6710",-65,"34:57:60:0f:67:10",6,36,0)
+CWLAP:(3,"GVT-D381",-88,"ec:22:80:99:d3:81",1,13,0)
+CWLAP:(4,"VIVO-5051",-45,"04:8d:39:08:50:50",10,10,0)
+CWLAP:(3,"NET AP 12",-85,"90:0d:cb:e6:17:a0",11,21,0)
+CWLAP:(3,"GVT-BBF1",-81,"68:15:90:b9:bb:f2",11,-7,0)

OK
AT+CWJAP="VIVO-5051","1604020062"
WIFI CONNECTED
WIFI GOT IP

OK
AT+CIFSR
+CIFSR:STAIP,"192.168.15.5"
+CIFSR:STAMAC,"60:01:94:2d:37:ec"

OK

```

Fonte: Autor.

No exemplo acima, foi utilizado os comandos AT para verificar primeiramente se a conexão estava correta, quando não há erros, o ESP enviará de volta um “OK”. Em seguida foi enviado o comando “AT+GMR” para verificar a versão do *firmware* instalado. O comando “AT+CWMODE?” é usado para ler ou configurar o modo de operação do módulo, onde 1 é o modo *STA*, 2 é *Acess Point* e 3 os dois modos juntos. “AT+CWLAP” listará todas as conexões sem fio possíveis para serem conectadas. “CW-JAP” é utilizado para se conectar a uma das redes disponíveis, para fazer isso é necessário escrever o nome da rede, em seguida a senha de acesso. Por último o comando “AT+CIFSR” retorna o *IP* do módulo.

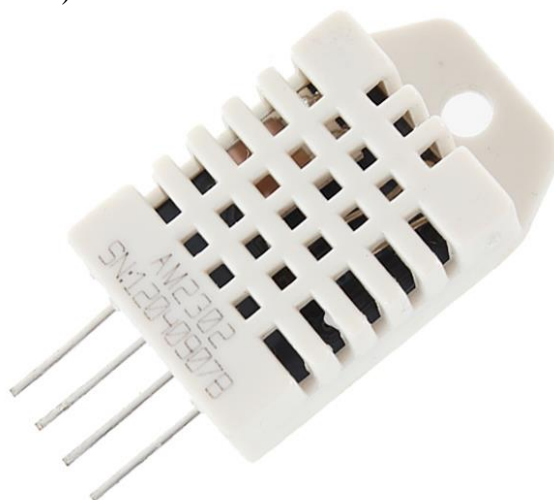
3.2 SENSOR DE TEMPERATURA E UMIDADE DHT22

O DHT22 é um sensor de temperatura e umidade que permite fazer leituras de temperaturas entre -40 a +80 graus Celsius e umidade entre 0 a 100%. Suas especificações podem ser consultadas através do *datasheet* que é uma folha com especificações e características técnicas de um produto.

Especificações:

- Comunicação I2C
- Tensão de operação: de 3 até 5 V
- Faixa de medição de umidade: 0 a 100% UR
- Faixa de medição de temperatura: -40° a +80°C
- Corrente: 2,5mA máxima durante uso, em *stand by* de 100 μ A a 150 μ A
- Precisão de umidade de medição: $\pm 2,0\%$ UR
- Precisão de medição de temperatura: $\pm 0,5$ °C
- Resolução: 0,1
- Tempo de resposta: 2 s

Figura 6 - DHT22 (FILIFELOP)



2.3 SENSOR DE TEMPERATURA DS18B20

O sensor DS18B20 é um sensor digital de temperatura com características interessantes para uso com o *Arduino*, que são o baixo custo e a facilidade de uso. Nesse projeto ele foi utilizado para medir a temperatura externa do ambiente.

Possui os seguintes atributos:

- Alimentação *DC* (*Direct Current*) entre 3.0V e 5.5V.
- Mede temperatura entre -55°C e +125°C.
- Não necessita de componentes externos.
- Resolução ajustável entre 9 e 12 bits.
- Comunicação *1-Wire*

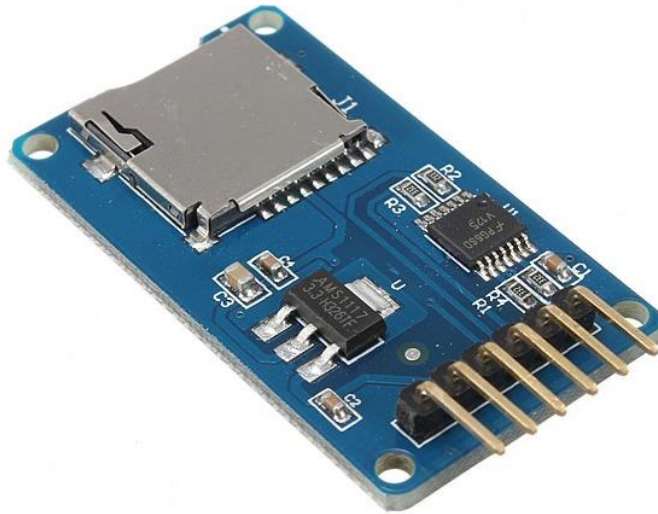
Figura 7 - Sensor DS18B20 (ARDUINOBR,2015)



2.4 MÓDULO DE CARTÃO MICRO-SD

Através desse módulo é possível realizar a escrita e leitura de um cartão SD, possui fácil ligação ao *Arduino*, suporta formato *FAT16*, *FAT32* e alimentação 3,3V ou 5,5V

Figura 8 - Módulo cartão micro SD (FILIPEFLOP,2016)



As especificações foram encontradas mais uma vez no *datasheet* do componente:

Especificações:

- Tensão de operação: 3,3 ou 5V
- Tensão de alimentação: 5V
- Interface SPI: MOSI, SCK, MISO e CS.
- Possui perfuração para fácil fixação

2.5 MÓDULO RTC DS3231

O módulo *Real Time Clock* (RTC) DS3231, é um relógio de tempo real de alta precisão e baixo consumo de energia, possui embutido um sensor de temperatura e um cristal oscilador para melhorar sua exatidão, também possui uma bateria inclusa para que caso haja uma queda de energia, os dados nele contido continuem salvos.

Este módulo é capaz de fornecer informação como segundo, minuto, dia, mês e ano, podendo operar tanto no formato de 12 como 24 horas.

Especificações

- Tensão de operação: 3,3-5V
- Chip: DS3231 (*datasheet*).
- Computa segundos, minutos, horas, dias da semana, dias do mês, meses e anos (de 2000 a 2099).
- Sensor de temperatura com ± 3 °C de exatidão.
- Chip de memória: AT24C32 (capacidade de 32kB que podem ser usadas como RAM estendida do micro controlador).

- Interface I2C – Barramento serial multimestre que é utilizado para conectar periféricos de baixa velocidade a uma placa mãe ou a um sistema embarcado.
- Circuito de detecção de falha de energia.
- Consome menos de 500nA no modo bateria com oscilador em funcionamento.
- Faixa de temperatura: 0 a 40°C.

Figura 9 - RTC DS3231 (FILIPEFLOP, 2016).



2.6 CABO MULTIVIAS

Para se ter um melhor manuseio dos sensores, utilizado para medições tanto internas quanto externas, foi utilizado um cabo multivias de 4 vias com blindagem.

Figura 10 - Cabo Multivias



Fonte: Autor.

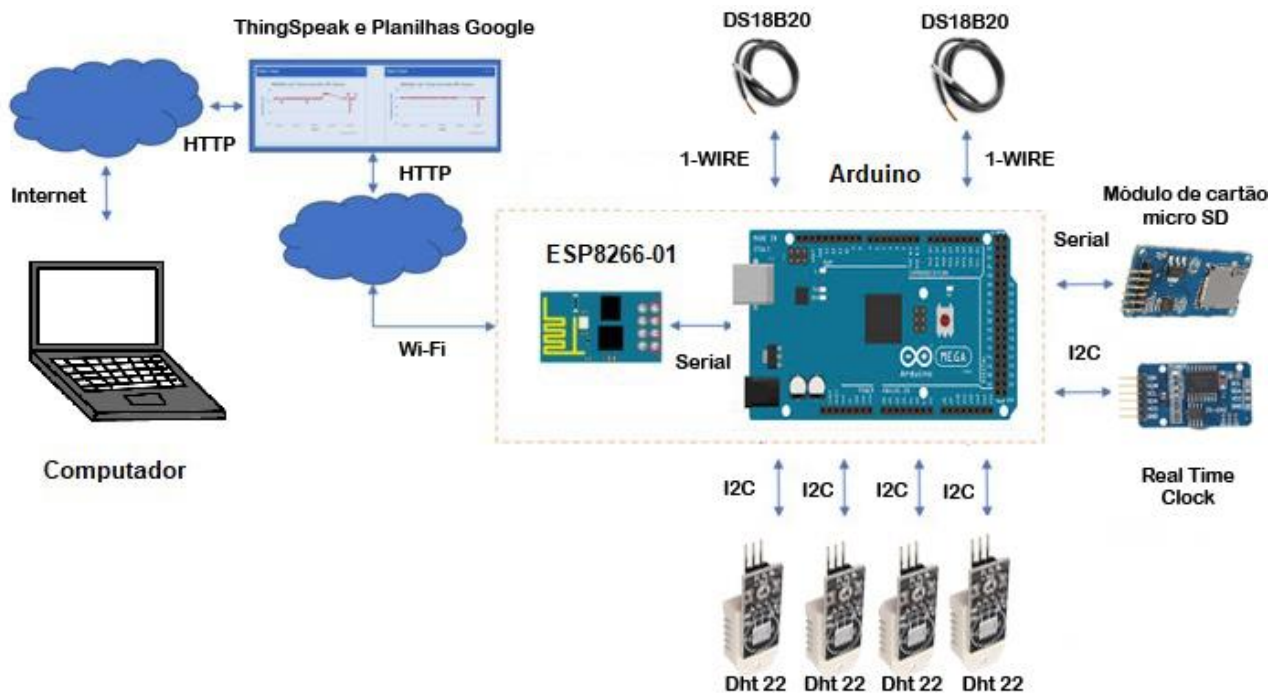
4. RESULTADOS

Neste capítulo são apresentados os resultados, desde a criação do *website*, fase de teste dos componentes, até o resultado final depois da instalação dos componentes na casa eficiente. As conexões ocorrem de duas maneiras, primeiramente o ESP realizará a conexão para as planilhas Google em seguida fará o mesmo processo com o *website ThingSpeak*, também é mostrado como é feito a conexão com a parte off-line do módulo de cartão SD.

4.1 ARQUITETURA GLOBAL DO SISTEMA

A arquitetura global do sistema pode ser vista na Figura 11. O protótipo é formado por 4 sensores de umidade e temperatura DHT 22, 2 sensores de temperatura DS18B20, 1 módulo de cartão micro SD, 1 *real time clock* e 1 ESP. Na Figura também é mostrado os protocolos de comunicação de cada módulo ou sensor com o *Arduino*. Na Tabela 1 é mostrado as conexões das entradas do *Arduino* com os pinos dos sensores.

Figura 11 - Arquitetura global do sistema



Fonte: Autor.

Tabela 1 - Conexão com o Arduino

Porta do Arduino	Pino do componente
10	Pino de dados do DS18B20
5	Pino de dados do DHT22 – Quarto frente
4	Pino de dados do DHT22 - Quarto do fundo
3	Pino de dados do DHT22 – Sala
2	Pino de dados do DHT22 - Cozinha
18 – TX	RX do ESP8266-01
19 – RX	TX do ESP8266-01
20 – SDA	SDA do RTC
21 - SCL	SCL do RTC
50	MISO do módulo de cartão micro SD
51	MOSI do módulo de cartão micro SD
52	SCK do módulo de cartão micro SD
53	CS do módulo de cartão micro SD

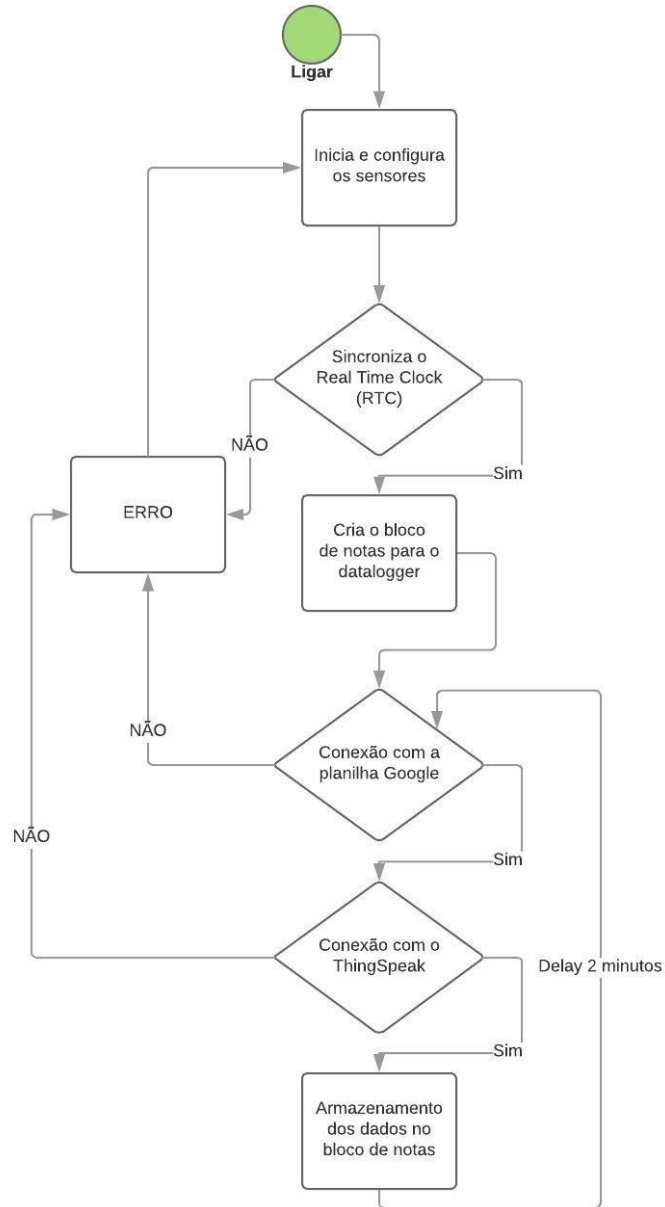
Fonte: Autor.

Após a energização do *Arduino*, os sensores de temperatura e umidade são configurados e iniciados automaticamente, em seguida ocorre a sincronização com o *RTC*, para saber qual a data e hora do momento em que é iniciado. Conforme a sincronização, será criado um arquivo-texto com o nome relativo ao dia e o mês informados pela sincronia do *RTC*. Em seguida é realizada a conexão *online*, utilizando o ESP, primeiramente conectando com o *site* da planilha *Google*, essa parte da conexão tem o intuito do armazenamento completo das informações, é neste *site* que se baseará o estudo das temperaturas. Em seguida a conexão com o *site ThingSpeak*, que ficará responsável por armazenar também as temperaturas, mas neste de forma não tão completa, pois os dados não ficam salvos por muito tempo (os dados ficam salvos por apenas três dias), o *website ThingSpeak* possui o intuito de gerar gráficos das temperaturas momentâneas, ficando em segundo plano com relação ao estudo completo. E por último, o armazenamento das variáveis no bloco de notas criado anteriormente seguido por um *delay* de aproximadamente 2 minutos.

O sistema é composto por duas partes, primeiramente a parte *online* através do ESP, e por segundo, a parte *off-line* através do módulo de cartão micro-SD. Ambas as partes utilizam

os mesmos sensores, ou seja, as medidas armazenadas *online* serão as mesmas medidas guardadas *off-line*. Para um melhor entendimento do programa, foi-se criado um fluxograma:

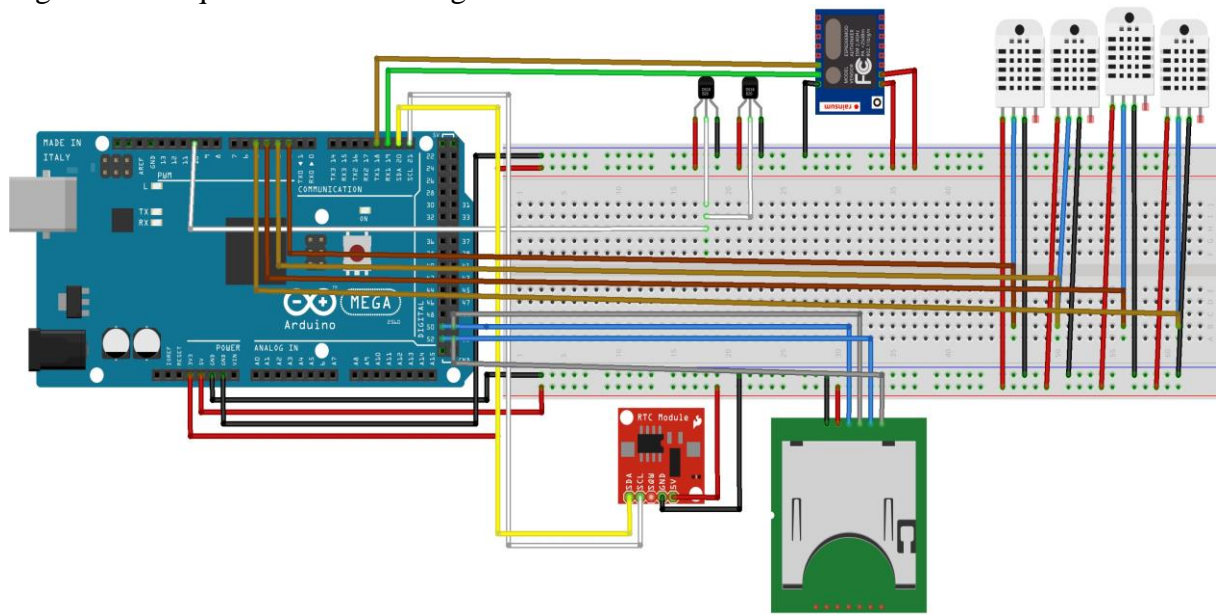
Figura 12 - Fluxograma



Fonte: Autor.

Com relação a montagem física, as conexões com o *Arduino* podem ser observadas na Figura 13.

Figura 13 - Esquemático da montagem

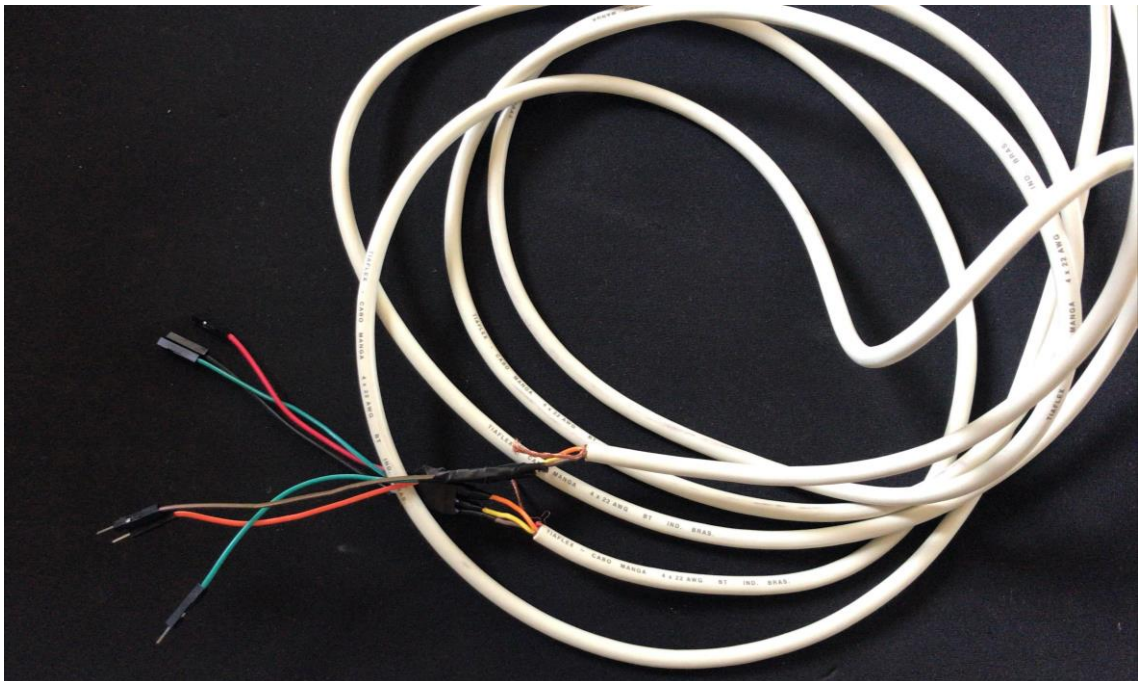


fritzing

Fonte: Autor.

Para a extensão da conexão entre os sensores e o *Arduino*, se utilizou cabos extensores multivias como o ilustrado na Figura 14. A Tabela 2 descreve o comprimento usado nas conexões com o cabo extensor multivias.

Figura 14 - Cabo Extensor Multivias



Fonte: Autor.

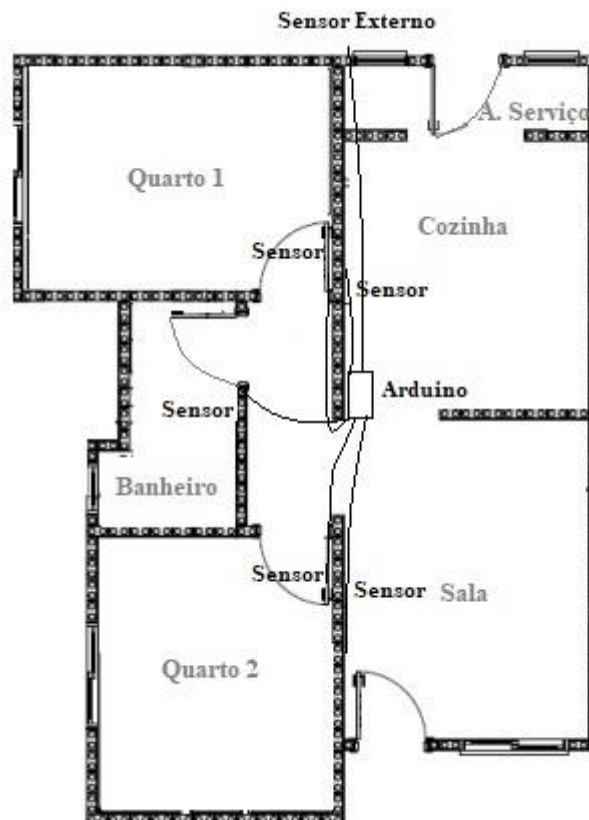
Tabela 2 - Comprimento dos cabos

Sensores	Comprimento
Cozinha	1 metro
Sala	3 metros
Quarto do fundo	3 metros
Quarto da frente	3 metros
Banheiro	2 metros
Externo	4 metros

Fonte: Autor.

O *Arduino* de onde parte-se todos os cabos estava fixado entre a cozinha e a sala, num ponto mais perto do centro da casa. Os sensores do quarto 1 (quarto do fundo) e quarto 2 (quarto da frente) se localizavam atrás da porta. O sensor do banheiro se encontrava ao lado da porta. Já os da sala e da cozinha estavam na metade do cômodo, também fixado na parede. Por último, o sensor externo foi colocado para fora da área de serviço. Para melhor entendimento foi mostrado na planta baixa da casa conforme a Figura 15.

Figura 15 - Planta baixa



Fonte: Autor.

4.2 CONFIGURAÇÕES DO SISTEMA AO THINGSPEAK

Para iniciar a conexão com o *ThingSpeak*, é necessário primeiramente realizar um cadastro no *Website*, para ter-se acesso a criação de um canal, nesse canal será colocado campos com os dados enviados via *Arduino*.

Durante a criação do canal, é onde pode-se dar um nome, descrição e habilitar o número de campos utilizados para envio dos dados, a interface pode ser visualizada na Figura 16.

Figura 16 – Criação do canal

New Channel

Nome

Descrição

Campo 1

Campo 2

Campo 3

Campo 4

Fonte: Autor.

A partir do canal criado, tem-se as opções de visualização, compartilhamento, configurações, chaves e exportação ou importação dos dados adquiridos, como pode ser visualizado na Figura 17.

Figura 17 - Informações do canal

ID do canal: **568514** | Teste com dht11

Author: [augustokuc](#)

Access: Públicos

Private View | **Public View** | Channel Settings | Sharing | Chaves | Data Import / Export

[+ Add Visualizations](#) | [+ Add Widgets](#) | [Data Export](#)

Channel Stats

Criada em: [24 days ago](#)

Updated: [24 days ago](#)

Last entry: [4 days ago](#)

Entries: 10529

Fonte: Autor.

Uma parte muito importante para o futuro código escrito na *IDE* do *Arduino* está nas chaves, lá se encontra uma chave de escrita no canal, que através dela, será feito o envio de informações para o *site*. Esta chave está sendo mostrada na Figura 18.

Figura 18 - Chave de escrita

Chave de Escrita

Chave

D29G5QSIL6X7L6NF

Gerar nova chave de escrita

Fonte: Autor.

Com a chave em mãos e a conexão estabelecida com o *modem*, pode-se começar a programar o código no *Arduino*. O protocolo de transporte utilizado será o tipo *TCP*, para realizá-la será utilizado os comandos “AT” já explicitado anteriormente neste trabalho.

Na Figura 19 é mostrada o código escrito na *IDE* do *Arduino*, primeiramente, será declarada uma *String* chamada “CMD”, que receberá s valores de “AT+CIPSTART” (responsável por estabelecer uma conexão), em seguida o tipo da conexão e o IP do *Website*, junto com a porta 80, responsável pelo serviço *HTTP*. Caso a conexão não esteja configurada corretamente, será relatado no monitor serial um erro.

Figura 19 - Código Arduino

```
// TCP connection
String cmd = "AT+CIPSTART=\\"TCP\\,\\"";
cmd += "184.106.153.149"; // api.thingspeak.com
cmd += "\",80";
SoftSer.println(cmd);
if (SoftSer.find("Error"))
{
  Serial.println("AT+CIPSTART error");
  return;
}
```

Fonte: Autor.

Após a realização da conexão, através do ESP será enviado uma mensagem, esta mensagem está contida em uma *string* nomeada “getStr”. Nela será colocada a chave de

escrita, que se encontra na Figura 20, junto com o comando “&field” responsável por identificar o campo que se deseja escrever, o código desta parte está na Figura 21.

Figura 20 - Código Arduino

```
String getStr = "GET /update?api_key=";
getStr += apiKey;
getStr += "&field1=";
getStr += String(strTemp);
getStr += "&field2=";
getStr += String(strTemp2);
getStr += "\r\n\r\n";
```

Fonte: Autor.

Para finalizar o envio da informação via ESP, mais uma vez será utilizado os comandos “AT”, neste caso será enviado o comando “AT+CIPSEND”, com o tamanho da mensagem criada anteriormente na *string*, como pode ser visualizado na Figura 21 com o código na IDE do *Arduino*.

Figura 21 - Código Arduino

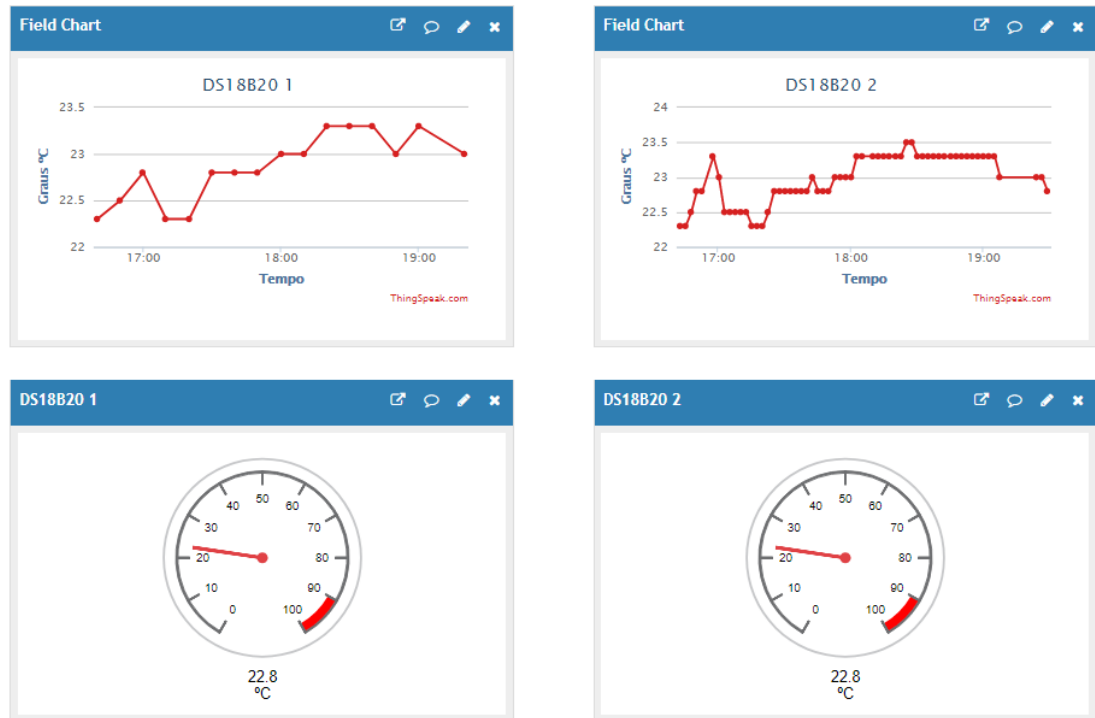
```
cmd = "AT+CIPSEND=";
cmd += String(getStr.length());
SoftSer.println(cmd);
if (SoftSer.find(">"))
{
  SoftSer.print(getStr);
}
else
{
  SoftSer.println("AT+CIPCLOSE");
}
```

Fonte: Autor.

Caso a mensagem não seja enviada, ou encontre algum problema no caminho, será escrito no monitor serial uma mensagem de erro.

O resultado de uma conexão com sucesso pode ser visualizado na Figura 22, através de gráficos que podem ser configurados com relação ao tempo e a amplitude, e também um medidor para informar o valor numérico do momento em que está sendo visualizado, lembrando que cada gráfico está interconectado através das habilitações dos campos como mostrado na Figura 16.

Figura 22 - Gráficos no website



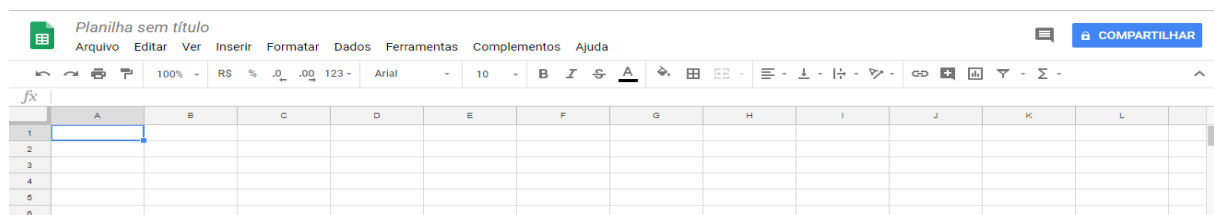
Fonte: Autor.

4.2 CONEXÃO COM A PLANILHA GOOGLE

A conexão com as planilhas do *Google*, são realizadas de uma maneira um pouco mais complexa, pois para ser concebida essa conexão, há a necessidade de ser empregada uma conexão segura (*SSL*), mas infelizmente, o *ESP-01* não possui capacidade computacional suficiente para tal requisito. Para que seja realizada uma conexão segura, será utilizado um *Website* chamado *PushingBox API*, este *site* lança um cenário de notificações que podem ser enviadas via requisição *HTTP*. Infelizmente o *site PushingBox* oferece um limite de 1000 execuções por dia, limitando assim a frequência do envio dos dados.

Inicialmente, é criado um documento parecido com o *Excel* via *Drive* no próprio site do *Google* chamado de *Planilha Google*, mostrado na Figura 23.

Figura 23 - Exemplo de Planilha



Fonte: Autor.

Por quesito de praticidade, a planilha foi configurada para a localidade do Reino Unido, devido as semelhanças com a utilização da escrita da data e as tabulações brasileiras como pode ser visualizado na Figura 24.

Figura 24 - Configuração da planilha

Configurações desta planilha

Geral
Cálculo

Localidade

Reino Unido ▾

Isto afeta os detalhes de formatação, como funções, datas e moeda.

Fuso horário

(GMT-03:00) Sao Paulo ▾

O histórico da sua planilha será gravado neste fuso horário. Este procedimento afetará todas as funções de tempo.

Idioma de exibição:
Português (Brasil)

Sempre usar os nomes de funções em inglês

Salvar definições

Cancelar

Fonte: Autor.

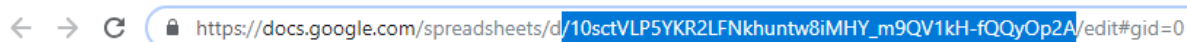
Através dessa planilha, pode-se utilizar um editor de *script*. Esse *script* é responsabilizado por realizar todos os comandos automaticamente dentro da planilha. O Script pode ser visto na Figura 25.

Figura 25 - Script da planilha

Fonte: Autor.

Para que o *script* funcione com sucesso, é necessário identificar qual o documento que será modificado através dele. Essa informação se encontra na *URL* do documento, como mostrado na Figura 26.

Figura 26 - URL do documento



Fonte: Autor.

Após identificado o endereço do documento, pode-se realizar a modificação do *script* para que ele funcione corretamente. Para tal tarefa, é copiado o endereço identificado anteriormente e é fixado na variável nomeada “*id*”, assim, é feita a programação para que o *script* identifique qual documento será modificado, o código utilizado no script está na Figura 27.

Figura 27 - Configuração script

```
var id = '10sctVLP5YKR2LFNkhuntw8iMHY_m9QV1kH-fQQyOp2A'; // Spreadsheet ID
var sheet = SpreadsheetApp.openById(id).getActiveSheet();
var newRow = sheet.getLastRow() + 1;
var rowData = [];
```

Fonte: Autor.

Próximo passo da configuração do *script* é a configuração de que tipo de informação será colocada e em qual coluna quando for executado. Como pode ser visto na Figura 28, as colunas são denominadas pelo nome “coluna_c”, “coluna_d” e assim por sequência, todas elas recebem um valor denominado “*value*”, que é configurado futuramente no próprio código do *Arduino*. Caso o valor colocado para ser escrito na coluna for um valor inválido, será resultado em um erro com a escrita “parâmetro não suportado”. Este código será executado em *looping* toda vez que o código do *Arduino* for executado, preenchendo todas as colunas de uma linha de uma só vez.

Figura 28 - Configuração do script

```

case 'coluna_c':
  rowData[2] = value;
  break;
  case 'coluna_d':
  rowData[3] = value;
  break;
  case 'coluna_e':
  rowData[4] = value;
  break;
  :
  :
  case 'coluna_k':
  rowData[10] = value;
  break;
default:
  result = "unsupported parameter";

```

Fonte: Autor.

Após realizada a programação correta do *script*, é necessário publica-lo *online* em modo público, para que qualquer pessoa possa modificar os parâmetros configurados anteriormente. Declarando como aplicação *online*, será gerado uma *URL* que será encarregada de modificar os dados mostrado na Figura 39.

Figura 29 - Declaração em modo público

Deploy as web app

Current web app URL: [Disable web app](#)

<https://script.google.com/macros/s/AKfyCbWKLcInekIDsH>

Test web app for your [latest code](#).

Project version:

16 ▼

Execute the app as:

Me (augikuki@gmail.com) ▼

You need to authorize the script before distributing the URL.

Who has access to the app:

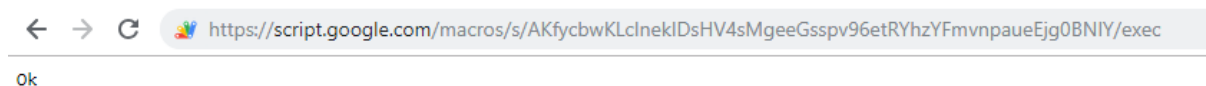
Anyone, even anonymous ▼

[Update](#) [Cancel](#) [Help](#)

Fonte: Autor.

Para se fazer um teste do funcionamento, é copiado a *URL* do campo “*Current web app URL*” e executado no navegador, caso seu funcionamento esteja correto, o resultado da execução será uma simples mensagem “ok”, como demonstrado abaixo na Figura 30:

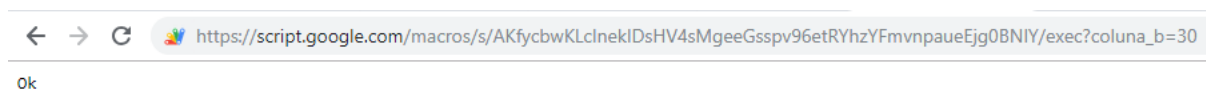
Figura 30 - URL gerada



Fonte: Autor.

Para que seja feita a alteração dos dados do documento, é necessário modificar o final desta *URL*, com um valor, como por exemplo “coluna_b=30” exibido na Figura 31.

Figura 31 - Modificação do parâmetro através da URL



Fonte: Autor.

Mais uma vez, caso seja executado corretamente, será exibido mais um OK, e será possível visualizar o valor escrito no documento, na Figura 32 é exibido o resultado.

Figura 32 - Demonstração do resultado

The image shows a Google Sheet interface. The spreadsheet has columns labeled A through J and rows labeled 1 through 6. The value "30" is entered in cell B1. The interface includes a menu bar with options like "Arquivo", "Editar", "Ver", "Inserir", "Formatar", "Dados", "Ferramentas", "Complementos", and "Ajuda". A status bar at the bottom indicates "Todas as alterações foram salvas no Google Drive".

	A	B	C	D	E	F	G	H	I	J
1		30								
2										
3										
4										
5										
6										

Fonte: Autor.

Partindo do princípio de ter-se realizado corretamente as configurações do *script*, pode-se começar o cadastro no *site PushingBox*, onde será adicionado um serviço de requisição de *html*. Na figura 33 é apontado que será adicionado um serviço de customizar a URL.

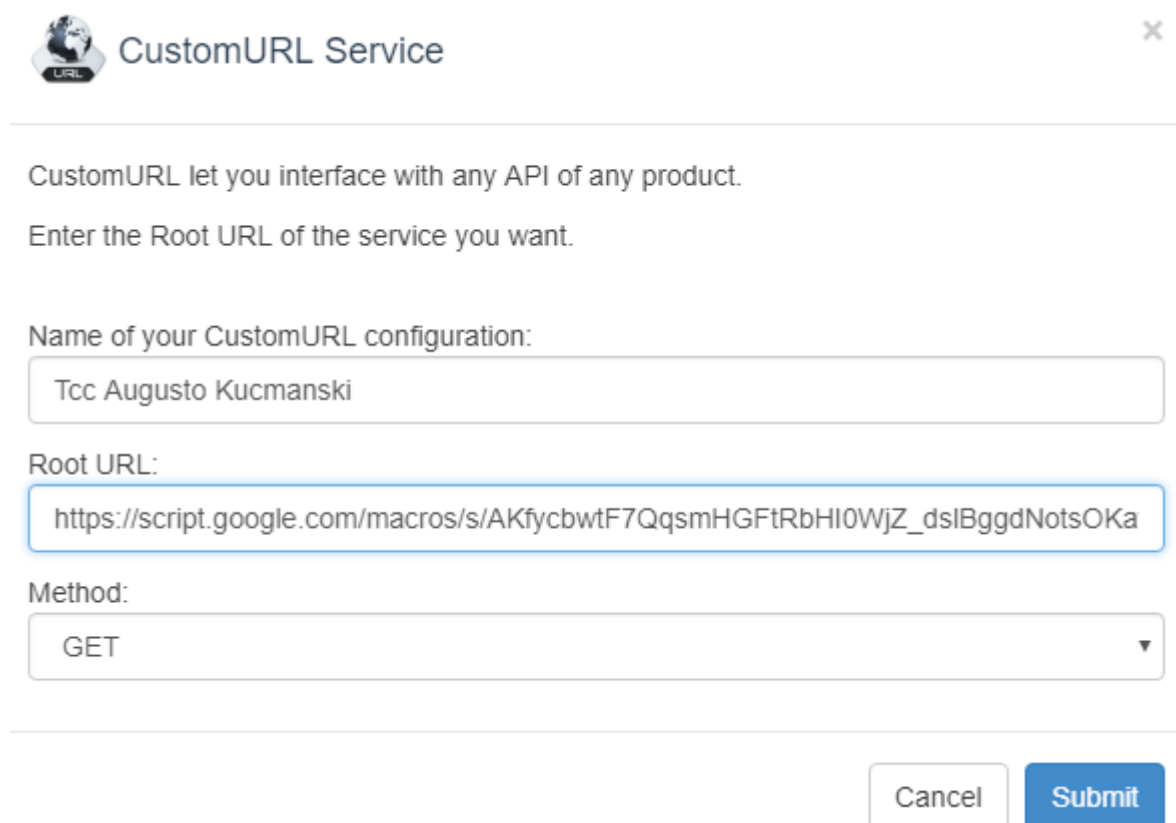
Figura 33 - Serviço de requisição



Fonte: Autor.

Neste serviço, será utilizado o endereço do *script* criado quando publicado em modo público, visualizado na Figura 34.

Figura 34 - Configuração do serviço de requisição



CustomURL let you interface with any API of any product.
Enter the Root URL of the service you want.

Name of your CustomURL configuration:
Tcc Augusto Kucmanski

Root URL:
https://script.google.com/macros/s/AKfycbwtF7QqsmHGfTRbHI0WjZ_dsIBggdNotsOKa

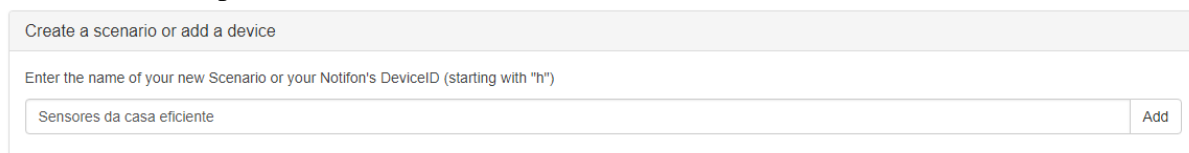
Method:
GET

Cancel Submit

Fonte: Autor.

Após criado o serviço, é possível adicionar dispositivos a ele, destacando na Figura 35 a criação do dispositivo.

Figura 35 - Criando dispositivo



Create a scenario or add a device

Enter the name of your new Scenario or your Notifon's DeviceID (starting with "h")

Sensores da casa eficiente Add

Fonte: Autor.

Depois de criado, tem-se a opção de adicionar uma ação ao dispositivo, como pode ser visto na Figura 36.

Figura 36 - Adicionando tarefa ao dispositivo



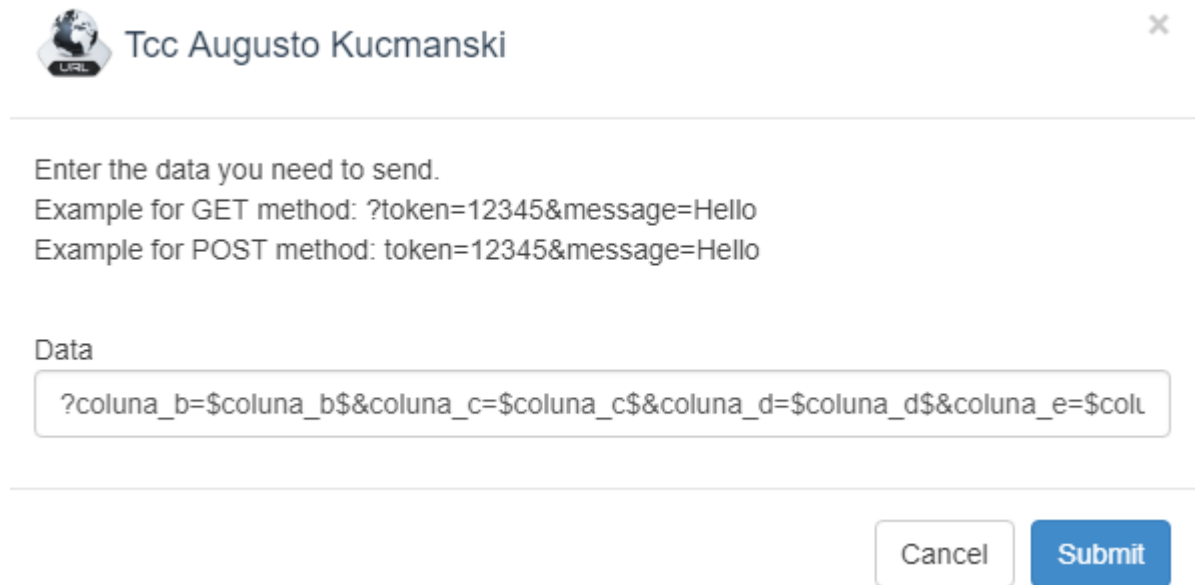
Tcc Augusto Kucmanski

Add an action with this service

Fonte: Autor.

O serviço adicionado será feito de uma maneira parecida com a utilizada no exemplo acima, utilizando os nomes das colunas programados no *script*, como por exemplo “coluna_b” recebendo um valor “?” visualizado na Figura 37.

Figura 37 - Configuração da tarefa



Enter the data you need to send.
 Example for GET method: ?token=12345&message=Hello
 Example for POST method: token=12345&message=Hello

Data

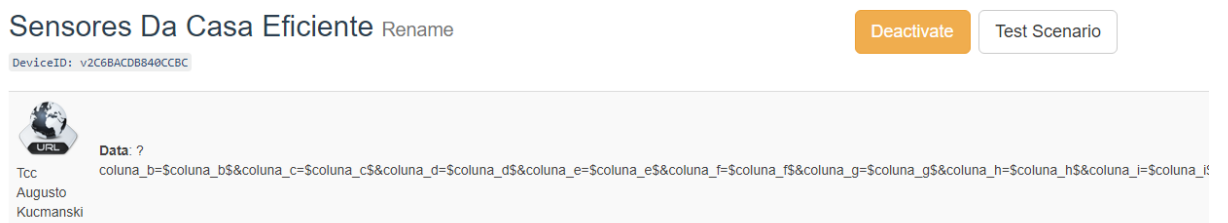
?coluna_b=\$coluna_b\$&coluna_c=\$coluna_c\$&coluna_d=\$coluna_d\$&coluna_e=\$coluna_e\$

Cancel Submit

Fonte: Autor.

Criada a ação do serviço, será gerado um *DeviceID*, que basicamente é uma chave de acesso, como já utilizado no caso anterior com o *website ThingSpeak*. Este *DeviceID* pode ser visto na Figura 38.

Figura 38 – DeviceID



Sensores Da Casa Eficiente Rename Deactivate Test Scenario

DeviceID: v2C6BACDB840CCBC

Data: ?coluna_b=\$coluna_b\$&coluna_c=\$coluna_c\$&coluna_d=\$coluna_d\$&coluna_e=\$coluna_e\$&coluna_f=\$coluna_f\$&coluna_g=\$coluna_g\$&coluna_h=\$coluna_h\$&coluna_i=\$coluna_i\$&coluna_j=\$coluna_j\$&coluna_k=\$coluna_k\$

Tcc Augusto Kucmanski

Fonte: Autor.

Através desse *DeviceID*, pode-se ter acesso a esse *API* enviando o requerimento “*GET*”, como no seguinte exemplo, onde é digitado o código no navegador:

“http://api.pushingbox.com/pushingbox?devid=v2C6BACDB840CCBC&coluna_b=1&coluna_c=2&coluna_d=3&coluna_e=4&coluna_f=5&coluna_g=6&coluna_h=7&coluna_i=8&coluna_j=9&coluna_k=10”. Ao utilizar este *link* no navegador, será executado o *script* através do Pushingbox, neste exemplo, é escrito valores de 1 a 10 nas colunas de B até K. O resultado do *script* é exposto na Figura 39.

Figura 39 - Resultado final

A	B	C	D	E	F	G	H	I	J	K
Data	Temperatura 1	Umidade 1	Temperatura 2	Umidade 2	Temperatura 3	Umidade 3	Temperatura 4	Umidade 4	Temperatura 5	Temperatura 6
08/09/2018 18:54	1	2	3	4	5	6	7	8	9	10

Fonte: Autor.

Após criado corretamente o *script* e configurado corretamente o *PushingBox*, pode-se testar a conexão utilizando o *Arduino*. Para estabelecer a conexão, é usado mais uma vez os comandos “AT”.

Neste caso, é utilizado o comando “AT+CIPSTART” (responsável por estabelecer uma conexão), em seguida o tipo da conexão e o *IP* do *Website*, junto com a porta 80, responsável pelo serviço *HTTP*. O código na IDE do *Arduino* pode ser visto na Figura 40.

Figura 40 - Código Arduino

```
String cmd = "AT+CIPSTART=\\"TCP\\",\\"";
cmd += DST_HOST;
cmd += "\\",80";

SoftSer.println(cmd); //send command to device
```

Fonte: Autor.

Para o envio da mensagem, é declarado uma *string* chamada “CMD”, essa *string* será responsável pelo comando “GET”, depois deste comando, é escrito o *DevID* criado pelo site *PushingBox*, após isso, é colocado os valores utilizados nas colunas da Planilha *Google*.

O resultado da *string* é a soma de todos os valores, como pode ser visualizado: “GET/pushingbox?devid=v2C6BACDB840CCBC&coluna_b=value&coluna_c=value”.

Através do comando escrito, será modificado no próprio *script* as linhas responsáveis por modificar os valores das colunas. Por fim, mais uma vez, é enviado o tamanho da mensagem para o ESP através do comando “AT+CIPSEND”, como pode ser checado na Figura 41.

Figura 41 - Código Arduino

```
String devid = "vB78B037B014FBC3";
cmd = "GET /pushingbox?devid=";
cmd += devid;
cmd += "scoluna_b=";
cmd += String(strTemp);
cmd += "scoluna_c=";
cmd += String(strTemp2);
cmd += " HTTP/1.1\r\n";
cmd += "Host: api.pushingbox.com\r\n\r\n";
SoftSer.print("AT+CIPSEND=");
SoftSer.println(cmd.length());
if (SoftSer.find(">"))
{
    SoftSer.println(cmd);
}
else
{
    SoftSer.println("AT+CIPCLOSE");
}
```

Fonte: Autor.

Tendo como resultado final uma tabela com os valores sendo enviados ciclicamente, transparecido na Figura 42.

Figura 42 - Resultado da planilha Google

	A	B	C	D	E	F	G	H	I	J	K
1	Data	Temperatura Cozinha	Umidade Cozinha	Temperatura Sala	Umidade Sala	Temperatura Quarto	Umidade Quarto	Temperatura Quarto 2	Umidade Quarto 2	Temperatura Externa	Banheiro
2	24/10/2018 18:00	24.3	50.5	24.4	55.1	24.1	54.5	24.8	50.9	25.3	25.3
3	24/10/2018 18:01	24.3	50.5	24.4	55.1	24.1	54.5	24.7	50.8	25.3	25.3
4	24/10/2018 18:03	24.4	50.5	24.5	55.2	24.1	54.7	24.7	50.9	25.3	25.3
5	24/10/2018 18:04	24.4	50.1	24.5	54.9	24.2	54.5	24.7	51	25.3	25.3
6	24/10/2018 18:06	24.5	49.8	24.5	54.6	24.1	54.3	24.8	50.9	25.3	25.3
7	24/10/2018 18:10	24.6	49.3	24.6	54.4	24.1	54.1	24.5	51	24.8	24.8
8	24/10/2018 18:11	24.4	49.8	24.4	54.7	24	54.4	24.2	51.4	24.5	24.5
9	24/10/2018 18:12	24.3	49.9	24.4	54.9	23.9	54.5	24.1	51.8	24.3	24.3
10	24/10/2018 18:14	24.2	50	24.2	55	23.8	54.6	23.9	52	24.3	24.3
11	24/10/2018 18:15	24.1	50.1	24.1	55.1	23.7	54.8	23.8	52.2	24	24
12	24/10/2018 18:17	24.1	50.5	24.1	55.4	23.6	55	23.7	52.5	24	24
13	24/10/2018 18:18	24	50.5	24	55.5	23.6	55.2	23.7	52.6	24	24
14	24/10/2018 18:20	24	50.6	24	55.7	23.6	55.2	23.7	52.7	24	24
15	24/10/2018 18:21	23.9	50.7	23.9	55.8	23.5	55.3	23.6	52.8	23.8	23.8
16	24/10/2018 18:23	23.9	50.8	23.9	55.9	23.5	55.4	23.6	52.9	23.8	23.8
17	24/10/2018 18:24	23.9	50.9	23.8	55.8	23.5	55.4	23.6	52.9	23.8	23.8
18	24/10/2018 18:25	23.9	51.1	23.8	56	23.5	55.6	23.6	53.1	23.8	23.8
19	24/10/2018 18:27	23.8	51	23.8	56	23.4	55.5	23.5	52.9	23.8	23.8
20	24/10/2018 18:28	23.8	51.1	23.8	56.2	23.4	55.7	23.5	53.1	23.8	23.8
21	24/10/2018 18:30	23.8	51.1	23.8	56.2	23.4	55.7	23.5	53.1	23.8	23.8
22	24/10/2018 18:31	23.8	51.2	23.8	56.3	23.4	55.8	23.5	53.2	23.8	23.8

Fonte: Autor.

4.3 CRIAÇÃO DO WEBSITE

Para a criação do *Website* foi utilizado um aplicativo de sistema de gerenciamento de conteúdo chamado *WordPress*, junto com um conjunto de *softwares* instalados via *Wamp*.

Para começar a criar o *Website*, é necessário realizar algumas configurações prévias, como a criação da base de dados e a instalação do *Wamp64*. Após feita sua instalação pode-se extrair a base de dados do *WordPress* na pasta onde foi instalado o *Wamp64*, evidente na Figura 43.

Figura 43 - Extração do WordPress no WAMP

Nome	Data de modific...	Tipo	Tamanho
pepolatumaini	07/02/2018 07:50	Pasta de arquivos	
sacana	04/01/2018 07:45	Pasta de arquivos	
test1	22/09/2018 14:36	Pasta de arquivos	
wamplangues	03/01/2018 17:19	Pasta de arquivos	
wampthemes	03/01/2018 17:19	Pasta de arquivos	
wordpress	22/09/2018 16:53	Pasta de arquivos	

Fonte: Autor.

Depois de ter-se extraído o conteúdo, é necessário acessar o seguinte endereço via navegador: “http://localhost/phpmyadmin”, para realizar o *login* com o usuário “root” e sem utilização de senha, indicado na Figura 44.

Figura 44 – Login

phpMyAdmin

Bemvindo ao phpMyAdmin

Língua - Language

Português - Portuguese

Entrada

Utilizador :

Palavra-passe:

Server Choice:

Executar

Fonte: Autor.

Quando é realizado o *login*, é possível ter acesso a configuração de uma nova base de dados inserida anteriormente, exibido na Figura 45.

Figura 45 - Configuração da criação da base de dados

phpMyAdmin

Current server: MySQL

Base de Dados

Criar base de dados

Nome da base de dados Agrupamento (Collation) Criar

Base de Dados	Agrupamento (Collation)	Acções
<input type="checkbox"/> information_schema	utf8_general_ci	Verificar Privilégios
<input type="checkbox"/> meuwordpress	latin1_swedish_ci	Verificar Privilégios
<input type="checkbox"/> mysql	latin1_swedish_ci	Verificar Privilégios
<input type="checkbox"/> pepolatumaini	latin1_swedish_ci	Verificar Privilégios
<input type="checkbox"/> performance_schema	utf8_general_ci	Verificar Privilégios
<input type="checkbox"/> sacana	latin1_swedish_ci	Verificar Privilégios
<input type="checkbox"/> sys	utf8_general_ci	Verificar Privilégios
<input type="checkbox"/> test1	latin1_swedish_ci	Verificar Privilégios
Totais: 8	latin1_swedish_ci	

Check all Com os seleccionados: Elimina

Fonte: Autor.

Para a criação da base de dados, é necessário apenas digitar um nome para sua identificação e clicar em na opção “Criar”. Este nome será importante para fazer a configuração correta do website.

Caso tudo esteja feito corretamente, pode-se começar a configuração do *Website*, acessando a página “localhost/wordpress” no navegador, o acesso é mostrado na Figura 46. Primeiramente com o nome do banco de dados criado anteriormente em seguida com o nome de usuário e senha do banco de dados.

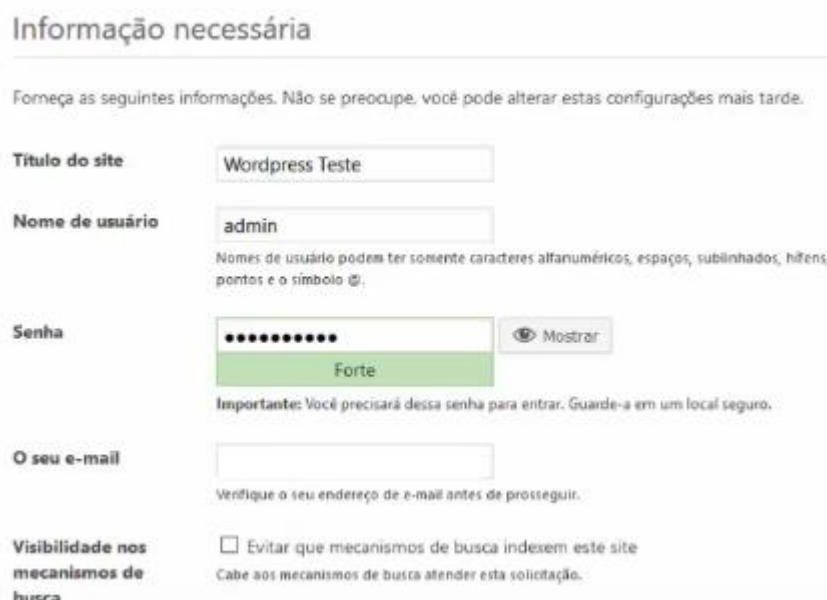
Figura 46 - Configurações do banco de dados



Fonte: Autor.

Feito isso, pode-se ir para o próximo passo, que é a criação do usuário administrador, necessário para ter os direitos de realizar as modificações na página, como pode ser visto na Figura 47.

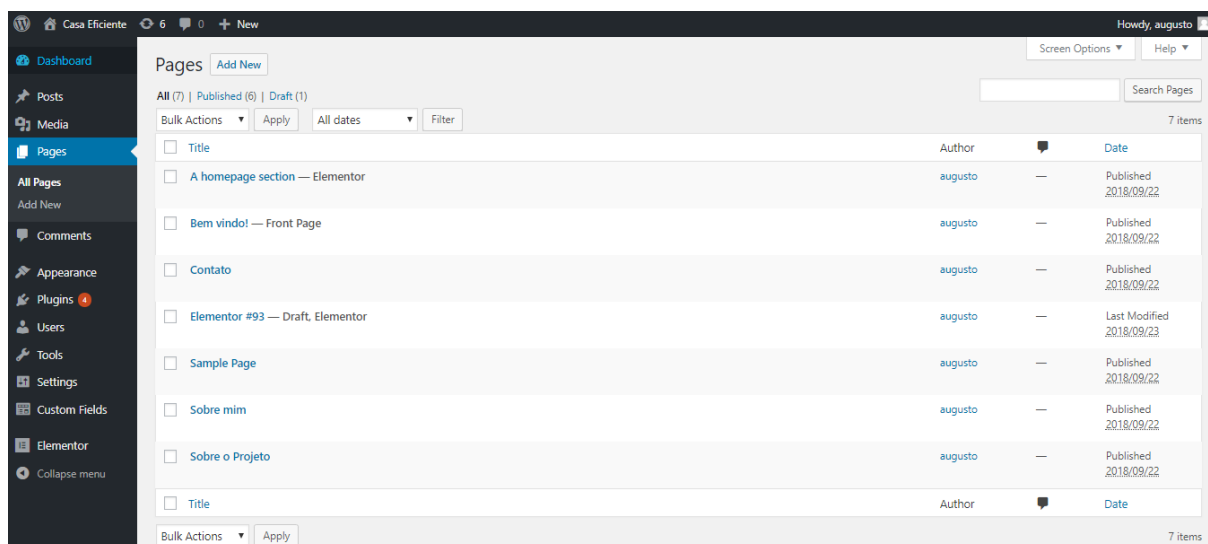
Figura 47 - Configuração do administrador



Fonte: Autor.

Após realizada as configurações prévias, pode-se começar a inserir conteúdo e alterações. A interface é demonstrada na Figura 48.

Figura 48 - Interface do WordPress

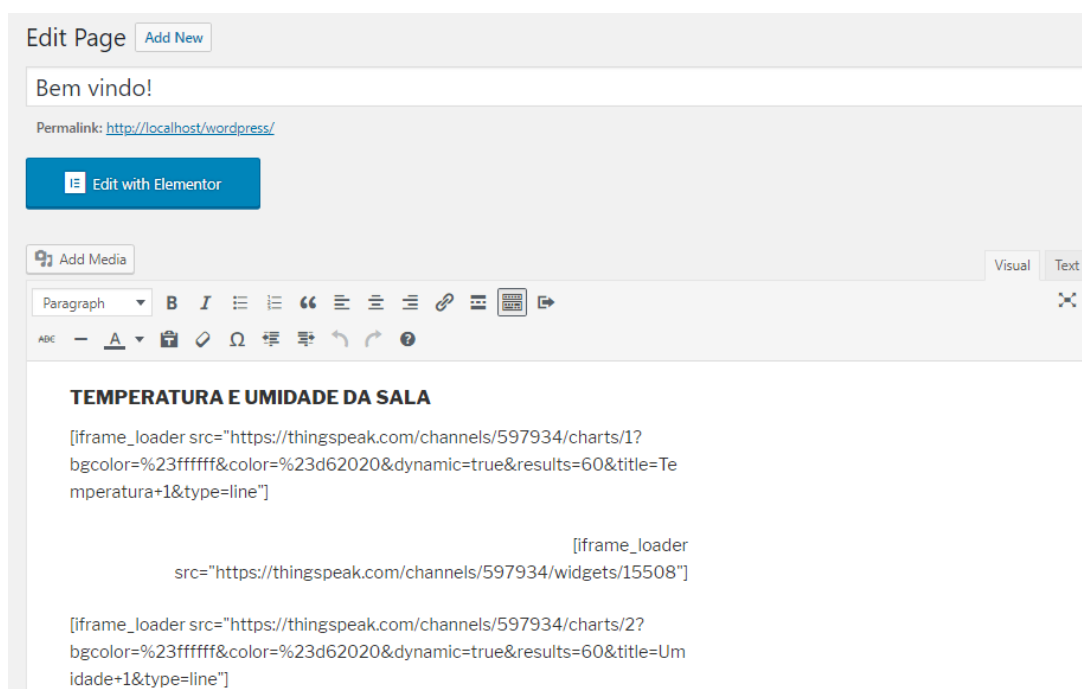


Fonte: Autor.

Neste trabalho foram criados três páginas principais “Contato”, “Sobre o projeto” e “Bem vindo!”. O *template* foi modificado com o logo da Universidade Federal de Santa maria e o logo da Engenharia de Controle e Automação.

Para inserir os gráficos criados no *ThingSpeak*, foi instalado um *Plugin* chamado *Widgets for Thingspeak*. Com este plugin, usa-se o comando `[i-frame_loader src=]`, como mostrado na Figura 49:

Figura 49 - Configuração da página inicial

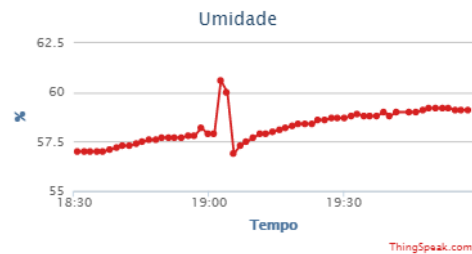
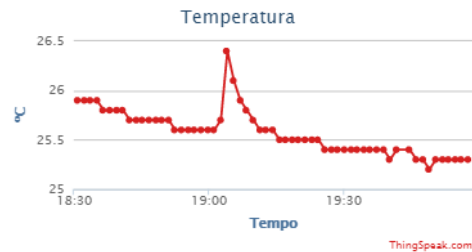


Fonte: Autor.

Tendo como resultado final a Figura 50:

Figura 50 - Resultado do website

TEMPERATURA E UMIDADE DA SALA



Fonte: Autor.

4.3.1 Colocando o website online

Primeiramente, para colocar um *site online*, é necessário a criação de um domínio. Existem diversos tipos de domínio, sejam pagos ou gratuitos, como o intuito desse projeto é ser de baixo custo, foi optado pela compra de um domínio no valor de 3,99 R\$. Este domínio foi encontrado através do *website* “<https://www.hostinger.com.br>”, que fez o serviço de descobrir quais domínios poderiam ser utilizados. O *site* é mostrado na Figura 51.

Figura 51 - Busca dos domínios

Fonte: Autor.

Optou-se por utilizar o domínio “casaeficiente.site” visualizado na Figura 52.

Figura 52 - Domínio utilizado

Fonte: Autor.

Para utilizar o domínio previamente adquirido, foi necessário realizar as mudanças dos nomes dos servidores, para que possa ser feito a hospedagem em outro servidor. Este outro *site* utilizado para a hospedagem é chamado *InfinityFree*, nele o serviço é feito de forma gratuita. A atualização dos servidores pode ser vista na Figura 53.

Figura 53 - Modificação dos servidores

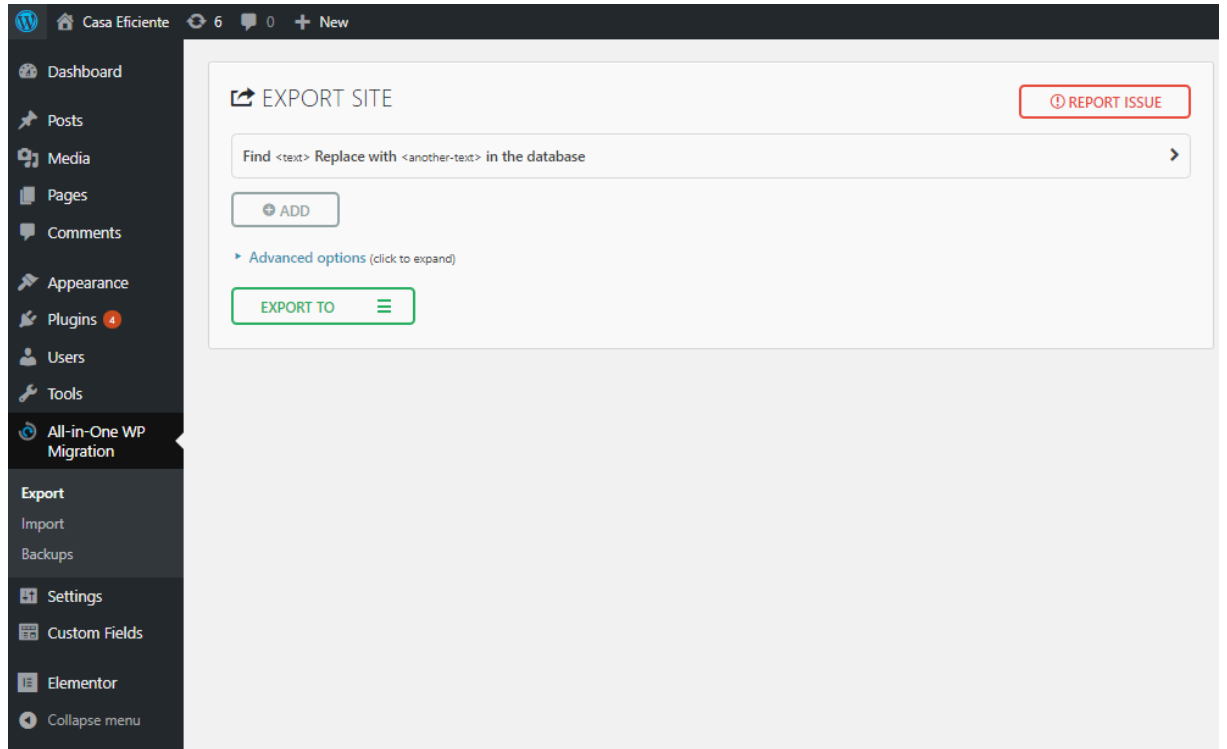
Atualizar nameservers ?

Nameserver #1	<input type="text" value="ns1.epizy.com"/>
Nameserver #2	<input type="text" value="ns2.epizy.com"/>
Nameserver #3	<input type="text"/>
Nameserver #4	<input type="text"/>

Fonte: Autor.

Com um *plugin* chamado *All-In-One Wp Migration*, fez-se um backup inteiro do *website* criado anteriormente de maneira *offline*, incluindo até mesmo o banco de dados. O processo é apresentado na Figura 54.

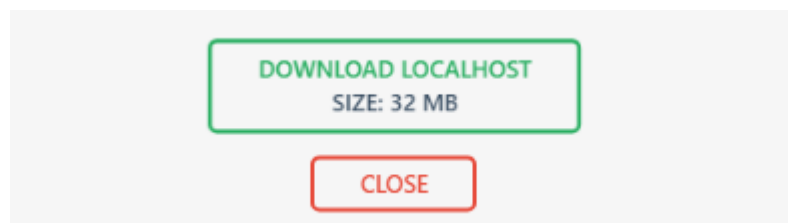
Figura 54 - Plugin All-in-One



Fonte: Autor.

Este *plugin* comprimirá tudo em um arquivo com o final “.wppress”, de 32 MB mostrado na Figura 55.

Figura 55 - Download do arquivo gerado



Fonte: Autor.

Com o domínio em mãos, o *backup* preparado, é feito então o cadastro em um *site* chamado *InfinityFree*, responsável pela reinstalação do *website* criado, mas agora utilizando o domínio adquirido. Detalhes da criação da conta pode ser observado na Figura 56.

Figura 56 - Criação da conta no site InfinityFree

Create a hosting account
Choose a subdomain or enter your own domain to create an account for. You can add more domains later.

Use a Free Subdomain

No configuration necessary. You can add more (sub)domains later..

Enter a free subdomain...

casaeficiente .rf.gd ▼

OR use your Own Domain

The domain needs to point to the nameservers: **ns1.epizy.com** and **ns2.epizy.com**. [Learn more.](#)

Enter your own domain...

Account Username
(generated automatically)

Account Password
(generated automatically)

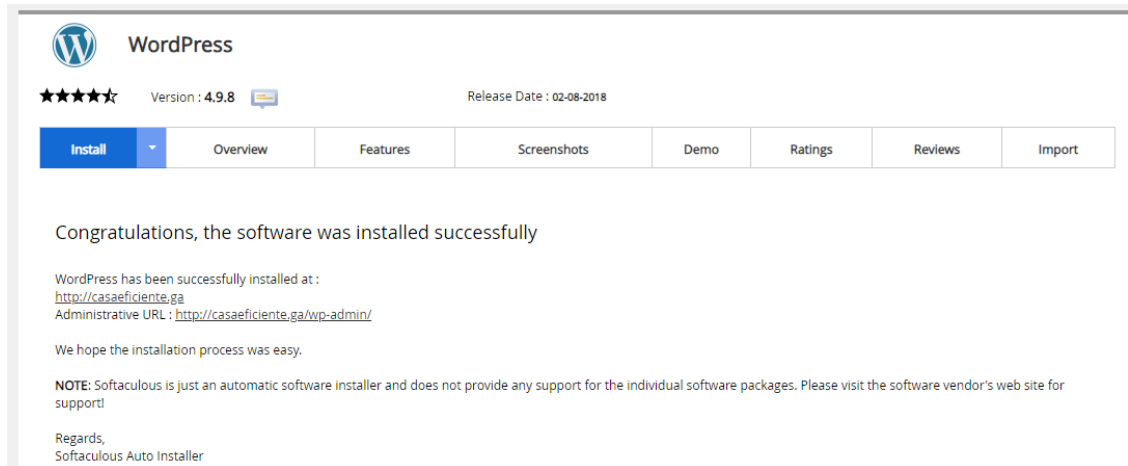
Account Details
Basic information about your account.

Username	epiz_22900113	
Label	Website for casaeficiente.ga	
Status	ACTIVE	
Password	VzAixEQgqNiqaJj	Show/Hide
Created on	2018-10-26	

Fonte: Autor.

Após realizada a criação correta da conta, com a configuração para ser utilizado o domínio adquirido anteriormente, é possível instalar a ferramenta do *WordPress*, destacado na Figura 57.

Figura 57 - Instalação do WordPress no novo site



Fonte: Autor.

A partir deste projeto em branco no *WordPress* recém instalado, foi utilizado mais uma vez o plugin *All-in-One*, responsável por fazer um *upload* do *backup* realizado anteriormente.

Por fim, é feito o acesso ao *website* através do domínio adquirido anteriormente para ter-se a certeza de que está funcionando sem nenhum problema, tendo como resultado a Figura 58.

Figura 58 - Acesso do website via domínio



Fonte: Autor.

4.4 PROGRAMAÇÃO DO DATALOGGER

Para a programação do *datalogger*, primeiro fez-se a sincronização do *RTC*, descrito na Figura 59:

Figura 59 - Sincronia do RTC

```
setSyncProvider(RTC.get);
Serial << F("Sincronizando com o RTC...");
if (timeStatus() != timeSet) Serial << F(" Falha!");
Serial << endl;
```

Fonte: Autor.

Em seguida, é feita a programação para que seja criado o documento onde ficará salvo as informações, mostrado na Figura 60:

Figura 60 - Código do Arduino para criação do documento

```
String diastring = String(day(t1));
String messtring = String(month(t1));
String text = ".txt";
String trav = "-";
String zero = "0";
String total = diastring + trav + messtring + text;
char chartotal[10]; //char com o numero 01-02.txt
total.toCharArray(chartotal, 10);
abre_arquivo_gravacao(chartotal);
```

Fonte: Autor.

O nome do arquivo é composto pelo dia e mês, com o final “.txt”, para que seja criado um bloco de notas, facilitando o acesso para todos os computadores. Após ter sido criado o bloco de notas, é possível gravar informações dentro dele, neste trabalho, será gravado ciclicamente a data, seguido da hora e as respectivas temperaturas. A primeira vez que o programa é executado, será impresso um cabeçalho composto por identificadores, como a data da hora em que é iniciado, junto com a temperatura e umidade dos respectivos cômodos.

Através da sincronia do *RTC*, pode-se utilizar as variáveis *day*, *month*, *year*, *hour*, *minute* e *second* para identificar dia, mês, ano, hora, minuto e segundo, o código para isto é mostrado na Figura 61. Cada vez que o valor das variáveis citadas anteriormente for menor que 10, é adicionado um 0 em sua frente, para que fique mais adequado ao modo como escrevemos as datas no dia a dia.

Figura 61 - Configuração da data e hora

```

file.print(day(t1));
file.print("/");
if (month(t1) < 10)
{
    file.print("0");
}
file.print(month(t1));
file.print("/");
file.print(year(t1));
file.print(" ");
if (hour(t1) < 10)
{
    file.print("0");
}
file.print(hour(t1));
file.print(":");
if (minute(t1) < 10)
{
    file.print("0");
}
file.print(minute(t1));
file.print(":");
if (second(t1) < 10)
{
    file.print("0");
}
file.print(second(t1));

```

Fonte: Autor.

Com relação ao armazenamento das variáveis de temperatura e umidade, pode se checar no código abaixo, lembrando que é colocado espaços entre as temperaturas para que fique de melhor entendimento quando gravado dentro do bloco de notas criado anteriormente, o código na IDE do *Arduino* é mostrado na Figura 62.

Figura 62 - Configuração temperatura e umidade

```

file.print(" ");
file.print(temp);
file.print(" ");
file.print(umid);
file.print(" ");
file.print(temp1);
file.print(" ");
file.print(umid1);
file.print(" ");
file.print(temp2);
file.print(" ");
file.print(umid2);

file.print(" ");
file.print(temp3);
file.print(" ");
file.print(umid3);
file.print(" ");
file.print(temp4);
file.print(" ");
file.print(temp5);
file.println(" ");

```

Fonte: Autor.

Após a execução do algoritmo escrito acima, tem-se o resultado primeiramente da criação do bloco de notas em seguida do armazenamento cíclico de informações (no caso o algoritmo foi executado dia 24/10), o resultado da criação pode ser visualizado nas Figura 63 e 64.

Figura 63 - Bloco de notas criado

24-10 24/10/2018 17:52 Documento de Texto

Fonte: Autor.

Figura 64 - Conteúdo do bloco de notas

Data	Hora	Temp_Cozinha	Umi_Cozinha	Temp_Sala	Umi_Sala	Temp_Quarto	Umi_Quarto	Temp_Quarto2	Umi_Quarto2	Temp_Externa	Temp_Banheiro
24/10/2018	16:51:49	24.20	51.40	23.80	55.50	23.60	55.70	23.70	54.80	24.00	24.00
24/10/2018	16:52:26	26.50	91.80	23.90	55.80	23.70	56.30	23.80	55.00	24.00	24.00
24/10/2018	16:53:02	28.70	92.30	24.00	55.80	23.80	56.60	23.90	55.00	24.00	24.00
24/10/2018	16:53:38	30.10	90.50	24.10	55.70	24.00	55.30	24.00	55.00	24.00	24.25
24/10/2018	16:54:15	29.90	44.50	24.20	55.60	24.10	55.20	24.10	54.70	24.00	24.25
24/10/2018	16:54:51	29.20	41.70	24.30	55.40	24.20	54.70	24.10	54.60	24.00	24.25
24/10/2018	16:55:27	28.40	42.40	24.30	55.10	24.30	54.60	24.20	54.40	24.00	24.25
24/10/2018	16:56:04	27.70	43.30	24.40	55.00	24.30	54.60	24.20	54.30	24.00	24.25
24/10/2018	16:56:41	27.20	44.40	24.40	54.70	24.40	54.70	24.30	53.90	24.00	24.25
24/10/2018	16:57:17	26.70	45.30	24.50	54.70	24.40	54.50	24.30	53.90	24.00	24.25
24/10/2018	16:57:53	26.30	46.20	24.50	54.60	24.40	54.50	24.40	53.70	24.00	24.25
24/10/2018	16:58:30	25.90	47.20	24.50	54.50	24.50	54.70	24.40	53.70	24.00	24.25
24/10/2018	16:59:06	25.70	48.00	24.60	54.50	24.50	54.70	24.40	53.60	24.00	24.25
24/10/2018	16:59:42	25.50	48.40	24.60	54.50	24.50	54.60	24.50	53.70	24.00	24.25
24/10/2018	17:00:20	25.30	48.70	24.60	54.40	24.50	54.40	24.50	53.50	24.25	24.25
24/10/2018	17:00:59	25.70	56.10	24.60	54.30	24.60	54.40	24.50	53.40	24.00	24.25
24/10/2018	17:01:38	26.20	48.80	24.70	54.20	24.60	54.30	24.50	53.50	24.00	24.25
24/10/2018	17:02:18	26.10	47.70	26.90	96.30	24.70	54.80	24.60	53.40	24.25	24.25
24/10/2018	17:02:56	25.90	47.50	28.30	99.50	24.70	54.70	24.60	53.40	24.25	24.25
24/10/2018	17:03:35	25.70	47.50	28.70	56.30	24.70	54.40	24.70	53.30	24.00	24.25
24/10/2018	17:04:14	25.60	48.00	28.20	49.40	24.80	54.40	24.70	53.00	24.00	24.25
24/10/2018	17:04:52	25.50	48.30	27.60	49.60	24.80	54.10	24.70	52.80	24.25	24.25
24/10/2018	17:05:28	25.50	49.10	27.90	52.30	25.20	59.10	24.80	51.60	24.25	24.25

Fonte: Autor.

Para que possa ser feita uma melhor análise dos dados coletados, pode-se utilizar *softwares* mais eficientes, como o *Excel*. Para isso, é necessário uma facilidade de importação para outros tipos de arquivos, como por exemplo uma planilha. Pensando se nisso, foram criados botões com macros, responsáveis pela importação dos dados dos blocos de notas criados anteriormente, já com mudança de tabulação e auto dimensionamento. Essas macros foram criadas inteiramente no programa Excel num ambiente chamado *Visual Basic for Applications (VBA)*. O código para importação dos dados está mostrado na Figura 65, junto com o ambiente de programação.

Figura 65 - Ambiente VBA

```

Sub Importar()
    Dim Arquivo As String

    With Application.FileDialog(msoFileDialogFilePicker)
        .Show
        Arquivo = .SelectedItems(1)
    End With

    Dim rg As Range
    Set rg = Range("A1")

    Open Arquivo For Input As #1

    Dim S As String, N As Integer, C As Integer, X As Variant
    Do Until EOF(1)
        Line Input #1, S
        C = 0
        X = Split(S, " ")
        For N = 0 To UBound(X)
            If X(N) <> "" Then
                rg.Offset(0, C) = X(N)
                C = C + 1
            End If
        Next N
        Set rg = rg.Offset(1, 0)
    Loop

    Close #1
End Sub

```

Fonte: Autor.

Após criado as macros, fez-se botões que ao serem clicados, são executadas as macros automaticamente, facilitando assim seu uso. Os botões foram denominados Importar e Dimensionar, mostrados na Figura 66.

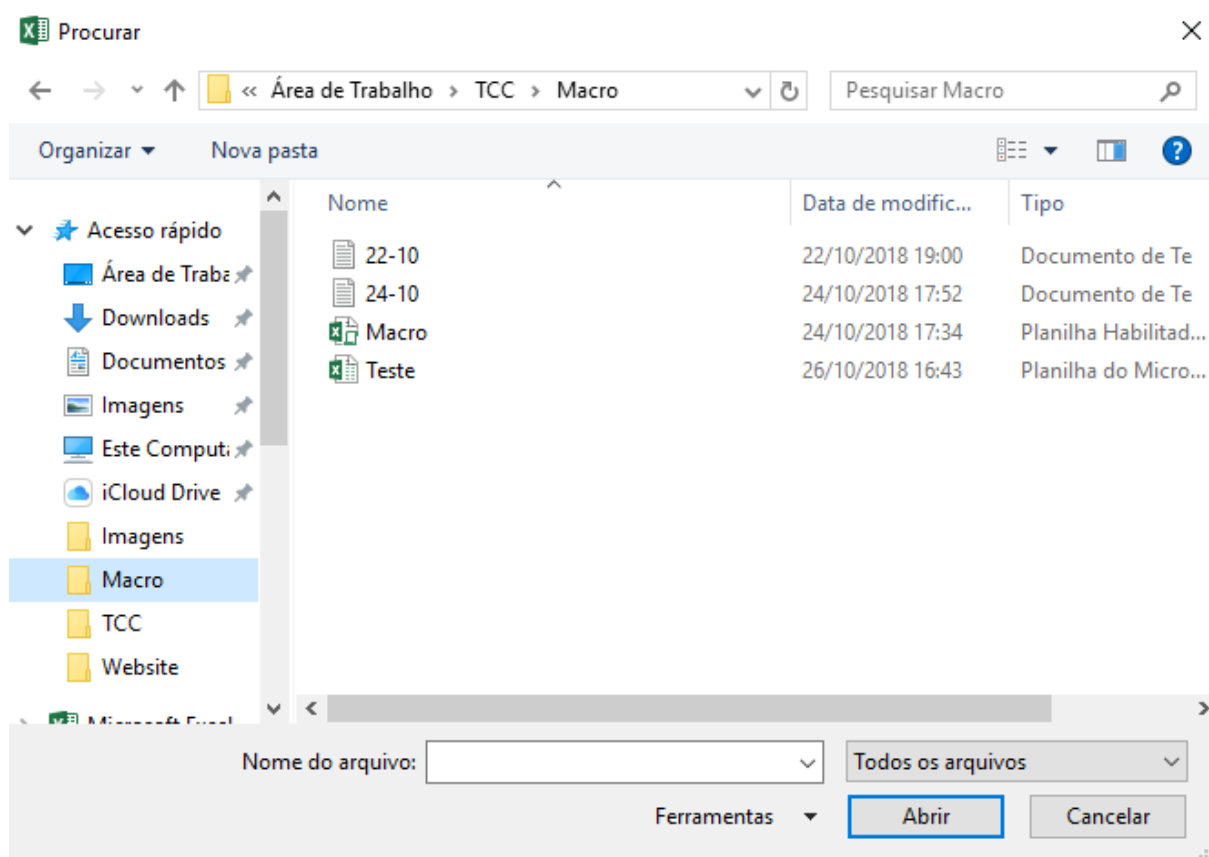
Figura 66 - Botões



Fonte: Autor.

Ao clicar no primeiro botão, será executado a *macro* de importar, abrindo uma nova janela para que seja selecionado o arquivo desejado, como resultado a Figura 67:

Figura 67 - Execução da primeira macro



Fonte: Autor.

Depois de aberto o arquivo, os dados serão importados com mudança automática na tabulação, o resultado é exposto na Figura 68.

Figura 68 - Resultado da importação

	A	B	C	D	E	F	G	H	I	J	K	L
1	Data	Hora	Temp_Cozinha	Umi_Cozinha	Temp_Sala	Umi_Sala	Temp_Quarto	Umi_Quarto	Temp_Quarto2	Umi_Quarto2	Temp_Externa	Temp_Banheiro
2	24/10/2018	16:51:49	24,2	51,4	23,8	55,5	23,6	55,7	23,7	54,8	24	24
3	24/10/2018	16:52:26	26,5	91,8	23,9	55,8	23,7	56,3	23,8	55	24	24
4	24/10/2018	16:53:02	28,7	92,3	24	55,8	23,8	56,6	23,9	55	24	24
5	24/10/2018	16:53:38	30,1	90,5	24,1	55,7	24	55,3	24	55	24	24,25
6	24/10/2018	16:54:15	29,9	44,5	24,2	55,6	24,1	55,2	24,1	54,7	24	24,25
7	24/10/2018	16:54:51	29,2	41,7	24,3	55,4	24,2	54,7	24,1	54,6	24	24,25
8	24/10/2018	16:55:27	28,4	42,4	24,3	55,1	24,3	54,6	24,2	54,4	24	24,25
9	24/10/2018	16:56:04	27,7	43,3	24,4	55	24,3	54,6	24,2	54,3	24	24,25
10	24/10/2018	16:56:41	27,2	44,4	24,4	54,7	24,4	54,7	24,3	53,9	24	24,25
11	24/10/2018	16:57:17	26,7	45,3	24,5	54,7	24,4	54,5	24,3	53,9	24	24,25
12	24/10/2018	16:57:53	26,3	46,2	24,5	54,6	24,4	54,5	24,4	53,7	24	24,25
13	24/10/2018	16:58:30	25,9	47,2	24,5	54,5	24,5	54,7	24,4	53,7	24	24,25
14	24/10/2018	16:59:06	25,7	48	24,6	54,5	24,5	54,7	24,4	53,6	24	24,25
15	24/10/2018	16:59:42	25,5	48,4	24,6	54,5	24,5	54,6	24,5	53,7	24	24,25
16	24/10/2018	17:00:20	25,3	48,7	24,6	54,4	24,5	54,4	24,5	53,5	24,25	24,25
17	24/10/2018	17:00:59	25,7	56,1	24,6	54,3	24,6	54,4	24,5	53,4	24	24,25
18	24/10/2018	17:01:38	26,2	48,8	24,7	54,2	24,6	54,3	24,5	53,5	24	24,25
19	24/10/2018	17:02:18	26,1	47,7	26,9	96,3	24,7	54,8	24,6	53,4	24,25	24,25
20	24/10/2018	17:02:56	25,9	47,5	28,3	99,5	24,7	54,7	24,6	53,4	24,25	24,25
21	24/10/2018	17:03:35	25,7	47,5	28,7	56,3	24,7	54,4	24,7	53,3	24	24,25

Fonte: Autor.

O segundo botão é responsável pela modificação do dimensionamento das linhas e colunas da tabela, caso já ocorra seu melhor dimensionamento possível, nada acontecerá.

4.6 RESULTADOS NA CASA EFICIENTE

As medições de temperatura e umidade na casa eficiente começaram às 15:00 no dia 06/11/2018, como pode ser visualizado na Figura 69, e terminou no dia 15/11/2018 às 07:44, como pode ser visualizado na Figura 70, resultando em 8057 medições. A frequência do armazenamento das informações varia de 1 para 2 minutos, isto devido ao limite de execuções que o *PushingBox* oferece.

Figura 69 - Resultados nas planilhas Google no início

Data	Temperatura Cozinha	Umidade Cozinha	Temperatura Sala	Umidade Sala	Temperatura Quarto	Umidade Quarto	Temperatura Quarto 2	Umidade Quarto 2	Temperatura Externa
06/11/2018 15:00	27.6	53.1	29.6	52.4	26.4	60.9	27.1	56.1	30.3
06/11/2018 15:01	27.3	53.6	29.5	52.2	26.4	60.7	27.2	55.1	29.5
06/11/2018 15:03	27.2	52.7	29.4	52.3	26.4	60.9	27.2	55.2	29.5
06/11/2018 15:04	27.1	52.5	29.3	50.4	26.3	60.6	27.1	55.2	29.8
06/11/2018 15:05	27	52.5	29.2	51.3	26.4	60.4	27.2	55.5	29.8
06/11/2018 15:07	26.9	53.3	29.2	52.4	26.3	60.6	27.1	56	29.8
06/11/2018 15:08	26.9	54	29.2	52.1	26.4	63	27.2	56.5	29.8
06/11/2018 15:10	26.9	54.3	29.2	52.2	26.4	61.2	27.3	56.2	29.8
06/11/2018 15:11	26.8	53.8	29.2	51.9	26.5	60.1	27.3	55.4	29.5
06/11/2018 15:13	26.8	53.3	29.2	51.2	26.5	62	27.3	56.1	29.3
06/11/2018 15:14	26.9	53	29.2	51.7	26.3	61.4	27.5	55.7	29.3
06/11/2018 15:16	26.9	52.7	29.2	51.3	26.3	61.4	27.5	56.1	29.5
06/11/2018 15:17	26.9	52.8	29.2	51.3	26.3	61.4	27.4	55.9	29.5
06/11/2018 15:19	26.9	52.9	29.2	50.5	26.4	61.4	27.3	56.1	29.8
06/11/2018 15:20	26.8	52.9	29.2	50.6	26.4	61.2	27.2	56.2	29.8
06/11/2018 15:21	26.8	52.8	29.1	50.2	26.4	61.3	27.2	56	29.8
06/11/2018 15:23	26.8	53.1	29.1	50.5	26.4	61.1	27.2	56.1	29.8
06/11/2018 15:24	26.8	53.3	29.2	50.4	26.4	61.4	27.2	56.5	29.8
06/11/2018 15:26	26.9	53.2	29.3	50.2	26.4	61.2	27.2	56.6	30

Fonte: Autor.

Figura 70 - Resultados nas planilhas Google no final

15/11/2018 07:16	19.6	59.4	20.1	59.4	21.5	59.3	20.7	57.7	16
15/11/2018 07:18	19.6	59.3	20.1	59.5	21.5	59.6	20.7	57.8	15.8
15/11/2018 07:19	19.6	59.4	20.1	59.5	21.4	59.5	20.7	58.1	16
15/11/2018 07:21	19.7	59.5	20.1	59.5	21.5	59.4	20.7	58	16
15/11/2018 07:22	19.7	59.4	20.2	59.6	21.6	59.5	20.7	58	16
15/11/2018 07:24	19.8	59.2	20.2	59.5	21.6	59.4	20.7	57.8	16
15/11/2018 07:25	19.8	59	20.2	59.5	21.6	59.7	20.7	57.7	16.3
15/11/2018 07:27	19.8	59.1	20.2	59.7	21.6	59.8	20.7	58.1	16.3
15/11/2018 07:28	19.8	59.4	20.2	59.7	21.8	59.4	20.7	58.3	16.3
15/11/2018 07:29	19.9	59.5	20.3	59.8	22	59	20.8	58.1	16.3
15/11/2018 07:31	19.9	59.6	20.3	59.8	22.1	58.7	20.9	57.9	16.5
15/11/2018 07:32	19.9	59.6	20.3	59.8	22.2	58.6	21	57.7	16.5
15/11/2018 07:34	19.9	59.5	20.3	59.8	22.2	58.6	21.1	57.6	16.8
15/11/2018 07:35	19.9	59.5	20.3	59.8	22.2	58.8	21.1	57.6	16.8
15/11/2018 07:37	20	59.4	20.3	59.7	22.3	58.4	21.1	57.2	16.8
15/11/2018 07:38	20	59.3	20.3	59.7	22.3	58.2	21.2	57.3	17
15/11/2018 07:40	20.1	59.1	20.4	59.7	22.4	58.2	21.2	57.2	17
15/11/2018 07:41	20.1	59	20.4	59.6	22.4	58.3	21.2	57.2	17.3
15/11/2018 07:42	20.2	59.2	20.5	59.6	22.4	58.1	21.3	57.1	17.5
15/11/2018 07:44	20.3	59.2	20.5	59.6	22.5	58	21.4	57.2	17.5

Fonte: Autor.

Para melhor visualização dos dados obtidos, foi plotado gráficos da temperatura em °C, com relação ao tempo em dias.

4.6.1 Resultados da cozinha

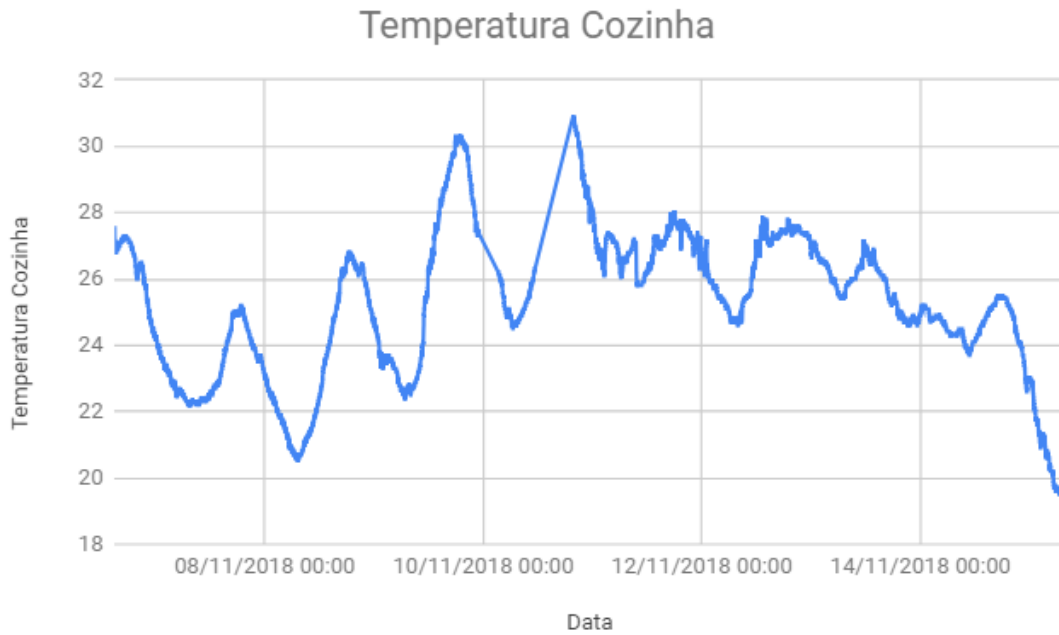
A Tabela 3 ilustra os dados obtidos na cozinha o que também pode ser visualizado nos gráficos da Figura 71 e 72.

Tabela 3 - Máximos e mínimos da cozinha

Temperatura máxima	30,9	10/11/2018
Temperatura mínima	19,5	15/11/2018
Umidade máxima	79,9	14/11/2018
Umidade mínima	38,7	10/11/2018

Fonte: Autor.

Figura 71 - Gráfico da temperatura da cozinha



Fonte: Autor.

Figura 72 - Gráfico da umidade da cozinha



Fonte: Autor.

4.6.2 Resultados da sala

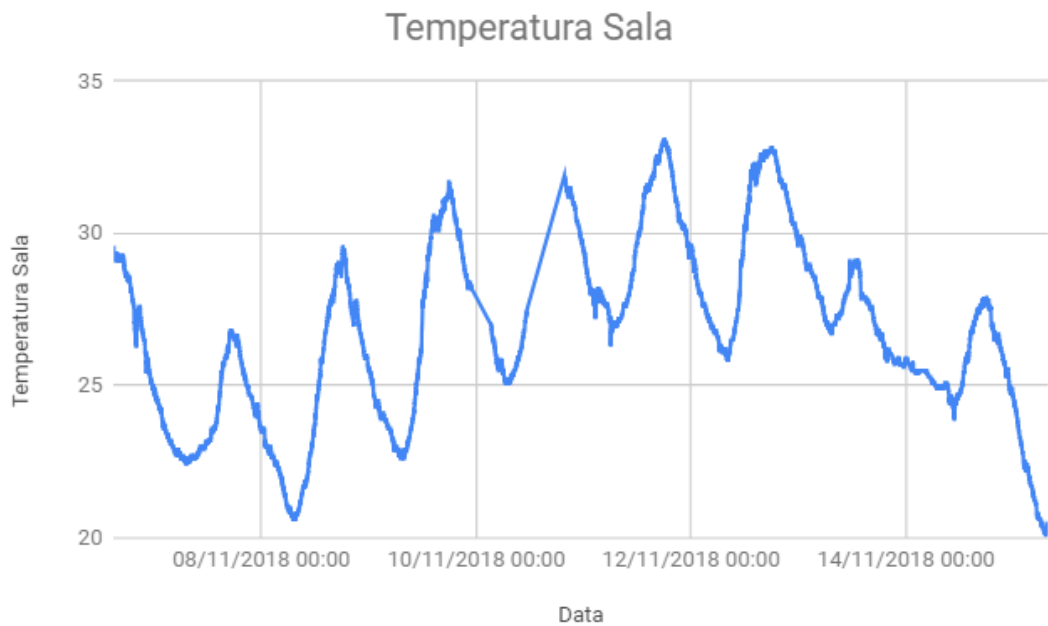
A Tabela 4 ilustra os dados obtidos na sala o que também pode ser visualizado nos gráficos da Figura 73 e 74.

Tabela 4 - Máximos e mínimos da sala

Temperatura máxima	33,1	11/11/2018
Temperatura mínima	20,1	15/11/2018
Umidade máxima	76,9	14/11/2018
Umidade mínima	39	14/11/2018

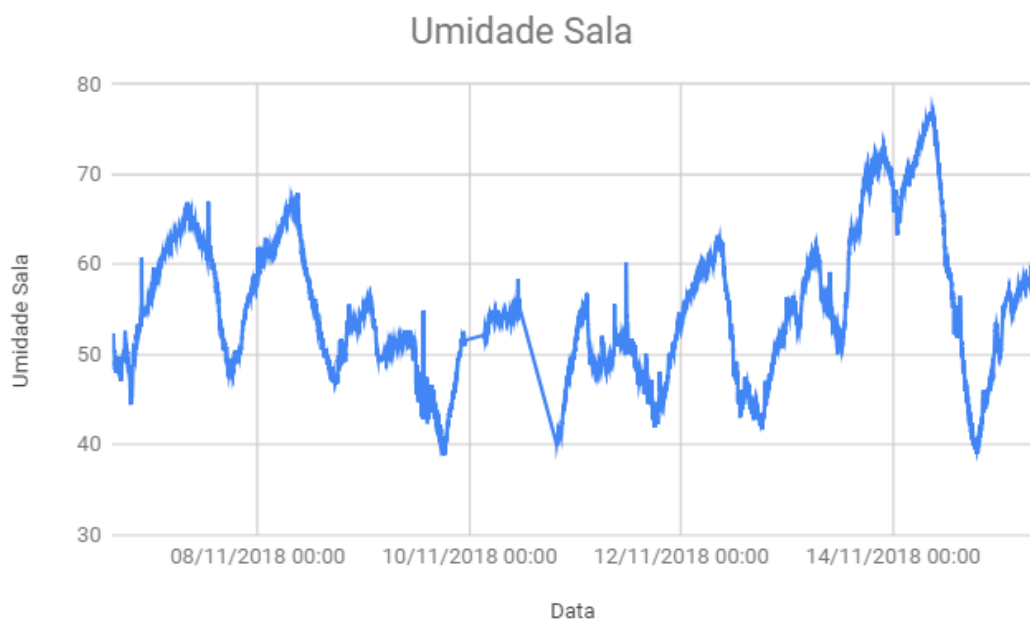
Fonte: Autor.

Figura 73 - Gráfico da temperatura da sala



Fonte: Autor.

Figura 74 - Gráfico da umidade da sala



Fonte: Autor.

4.6.3 Resultado do quarto dos fundos

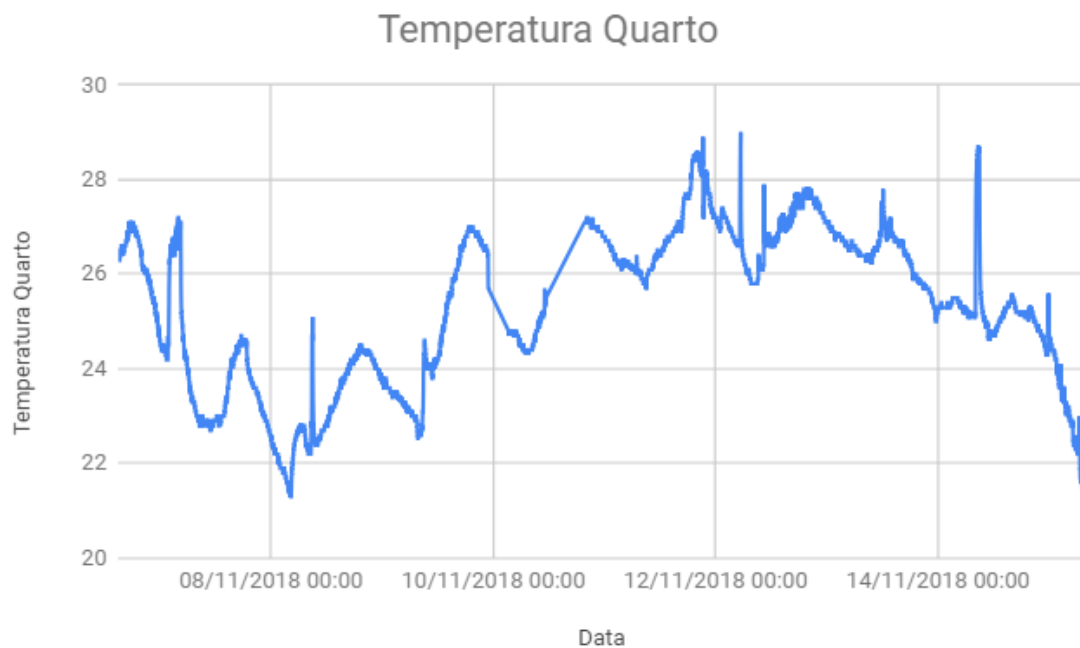
A Tabela 5 ilustra os dados obtidos no quarto dos fundos o que também pode ser visualizado nos gráficos da Figura 75 e 76. Foi detectado algum distúrbio devido mal contato do sensor no dia 15/11, que foi descartado.

Tabela 5 - Máximos e mínimos do quarto dos fundos

Temperatura máxima	28,9	11/11/2018
Temperatura mínima	21	15/11/2018
Umidade máxima	77,9	14/11/2018
Umidade mínima	18,2	09/11/2018

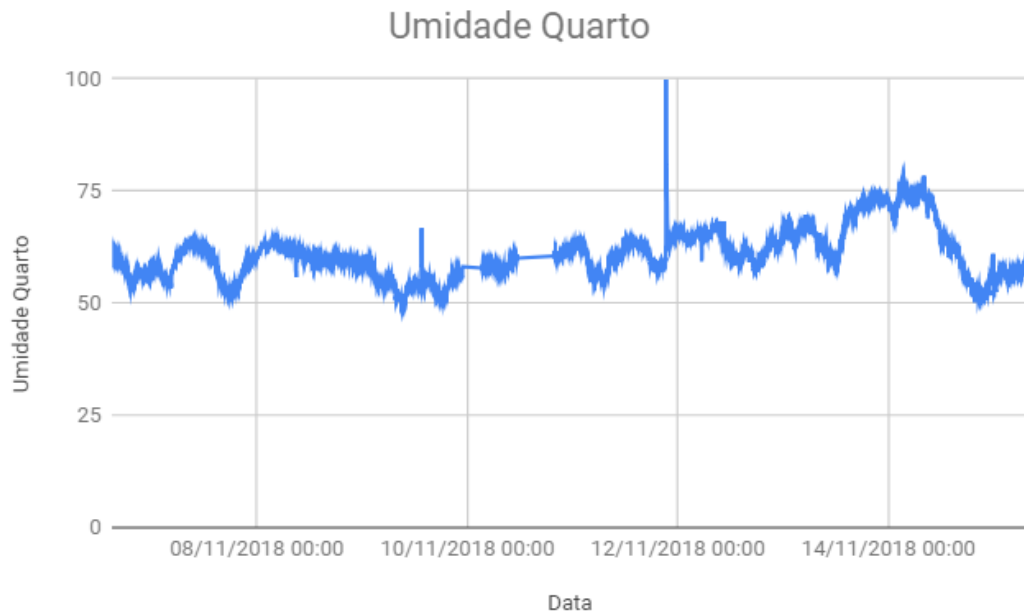
Fonte: Autor.

Figura 75 - Gráfico da temperatura do quarto dos fundos



Fonte: Autor.

Figura 76 - Gráfico da umidade do quarto dos fundos



Fonte: Autor.

4.6.4 Resultado do quarto da frente

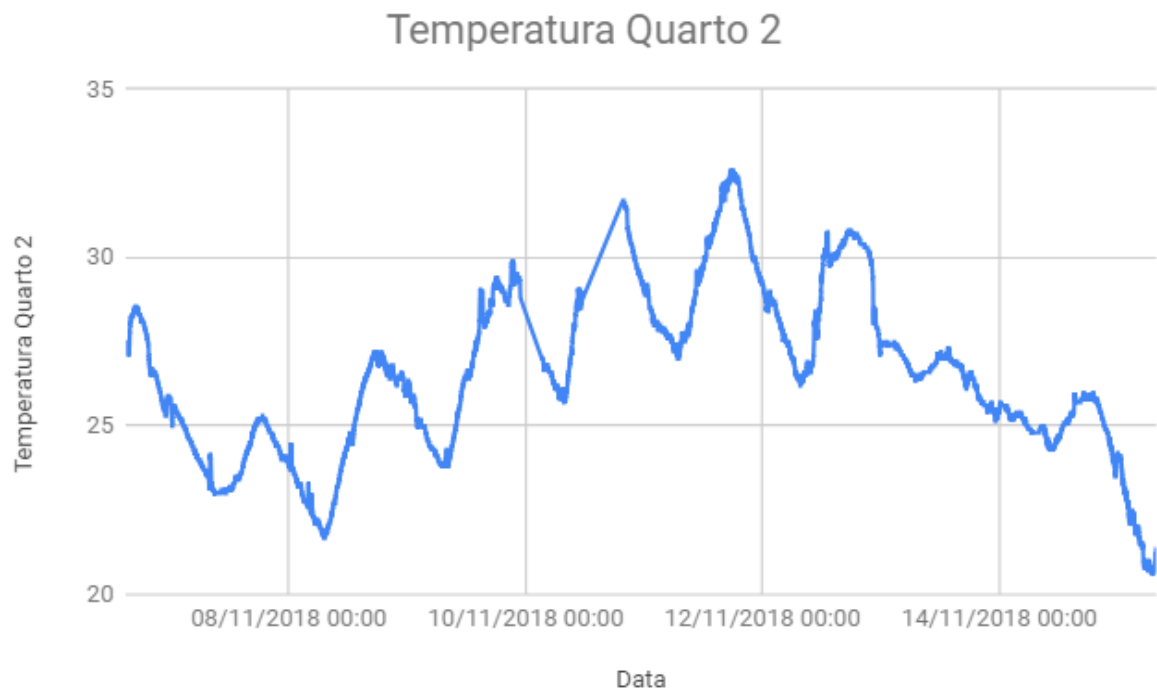
A Tabela 6 ilustra os dados obtidos no quarto da frente o que também pode ser visualizado nos gráficos da Figura 77 e 78.

Tabela 6 - Máximos e mínimos do quarto da frente

Temperatura máxima	32,6	11/11/2018
Temperatura mínima	20,6	15/11/2018
Umidade máxima	77,7	14/11/2018
Umidade mínima	38,5	11/11/2018

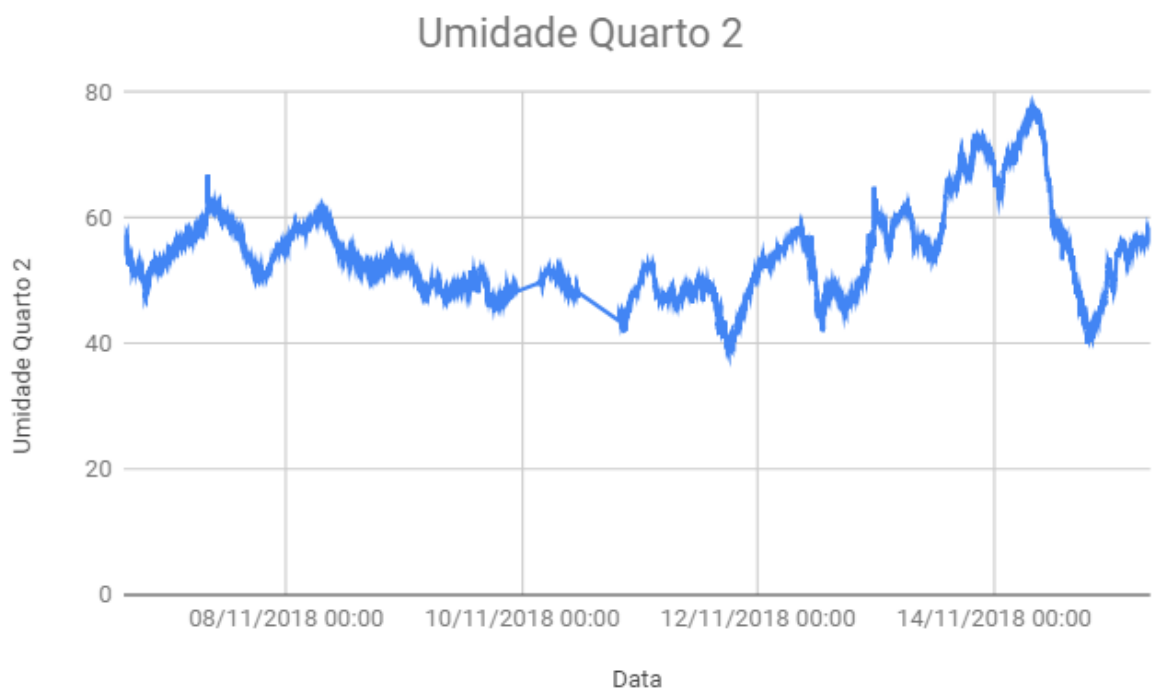
Fonte: Autor.

Figura 77 - Gráfico da temperatura do quarto da frente



Fonte: Autor.

Figura 78 - Gráfico da umidade do quarto da frente



Fonte: Autor.

4.6.5 Resultados da temperatura externa

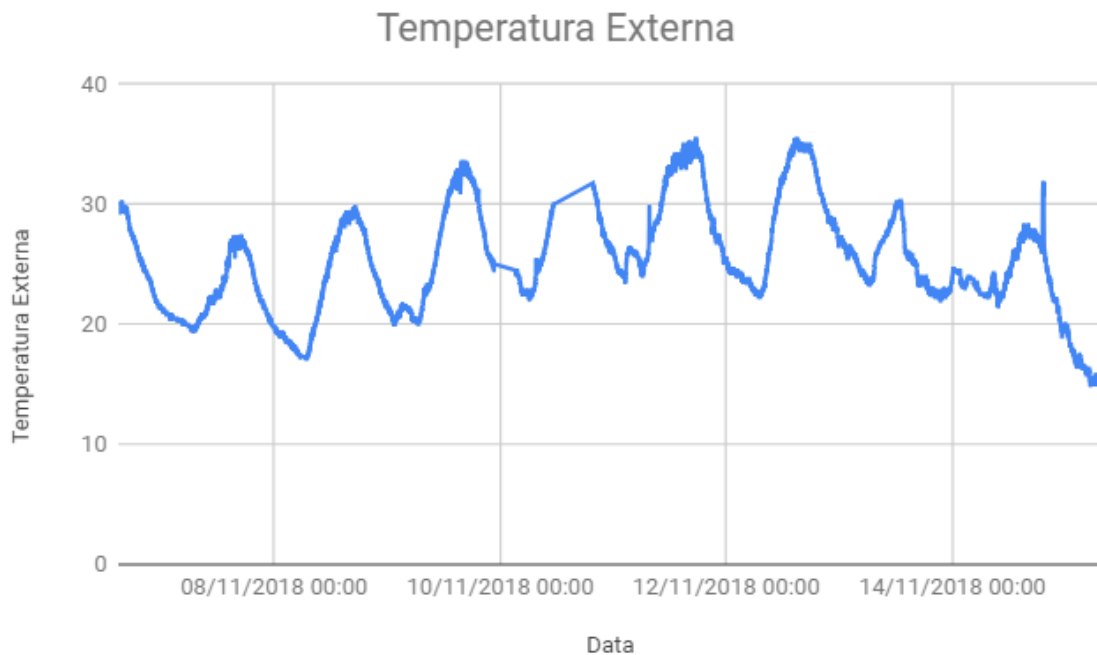
Os resultados externos foram focados em temperatura, devido ao fato do sensor utilizado ser a prova de água, enquanto o sensor de umidade não cumpre este requisito, tornando-o inviável para a utilização. A Tabela 7 e a Figura 79 ilustram os resultados.

Tabela 7 - Máximos e mínimos da temperatura externa

Temperatura máxima	35,5	11/11/2018
Temperatura mínima	14,8	15/11/2018

Fonte: Autor.

Figura 79 - Gráfico da temperatura externa

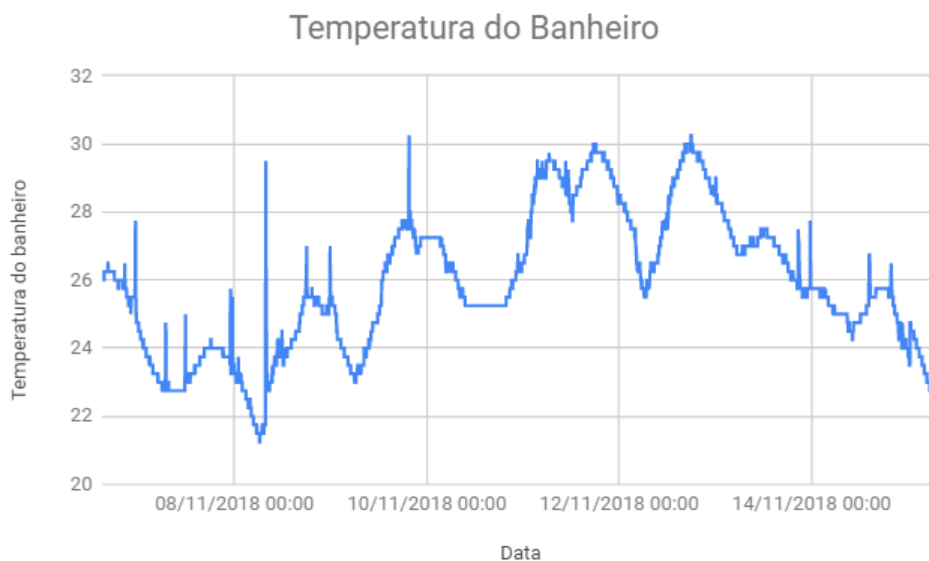


Fonte: Autor.

4.6.6 Resultados do banheiro

Os resultados do banheiro não foram obtidos de forma satisfatória, como de possível análise no gráfico da Figura 80, os valores acabam por oscilar com muita frequência, tornando-o ineficiente para o estudo neste trabalho. Isto se deve provavelmente aos banhos que alteram as características medidas.

Figura 80 - Temperatura do banheiro



Fonte: Autor.

4.7 COMPARAÇÃO COM RESULTADOS ANTERIORES

Para efeito de comparação serão confrontados resultados do estudo chamado prévio realizado por Joaquim César Pizzutti dos Santos chamado de Avaliação Térmica de Casa Popular Eficiente na Zona Bioclimática 2 Brasileira no Período de Verão (SANTOS et al.,2015). Neste estudo foram obtidos os resultados com relação a temperaturas máximas e mínimas com ventilação aberta estão representados na Tabela 8. Vale lembrar que tanto o estudo de 2015 quanto o estudo de 2018 foi-se feito na mesma residência, com diferença apenas na localização dos sensores nos cômodos da casa.

Tabela 8 - Temperaturas máximas e mínimas de 2015

Ambiente	Data	Temperatura Máxima (°C)	Temperatura Mínima (°C)	Amplitude (°C)
Cozinha	07/02/2015	37,660	28,310	9,350
Sala	07/02/2015	37,660	28,505	9,155
Quarto dos fundos	07/02/2015	36,565	29,100	7,465
Quarto da frente	07/02/2015	35,060	28,310	6,750
Externa	07/02/2015	39,900	24,015	15,885

Fonte: SANTOS et al.,2015.

As temperaturas mais elevadas foram registradas nos cômodos da cozinha e da sala, por estarem posicionados com suas paredes externas voltadas ao norte e oeste. Já o quarto do fundo é o cômodo com a menor temperatura máxima interna medida, pois a posição geográfica desse quarto possui suas paredes externas voltadas ao sul e leste influenciando positivamente no período quente.

Com relação ao estudo realizado em 2018 por este trabalho se escolheu o dia com a maior temperatura para fazer uma tabela com a amplitude de sua variação, como pode ser analisado na Tabela 9.

Tabela 9 - Tabela temperaturas de 2018

Ambiente	Data	Temperatura Máxima (°C)	Temperatura Mínima (°C)	Amplitude (°C)
Cozinha	11/11/2018	28	25,8	2,2
Sala	11/11/2018	33,1	26,3	6,8
Quarto dos fundos	11/11/2018	28,9	25,7	3,2
Quarto da frente	11/11/2018	32,6	27	5,6
Externa	11/11/2018	35,5	24	11,5

Fonte: Autor.

Pode-se perceber que os ambientes que mais possuem uma variação térmica são a sala e o quarto da frente. Já o dormitório do fundo é o ambiente com a menor temperatura máxima interna. Tanto a temperatura do cômodo dos fundos quanto a da sala se dá mais uma vez devido as posições geográficas que os cômodos se encontram. Com relação ao quarto frontal, este possui uma temperatura mais elevada devido provavelmente a um projeto que foi realizado anteriormente que se baseia no aquecimento do quarto através de um sistema de calefação que é fundamentado na transferência de temperatura de pedras que são aquecidas pelo sol para o quarto, como pode ser visto nas Figuras 81 e 82.

Figura 81 - Sistema externo de calefação



Fonte: Autor.

Figura 82 - Sistema interno de calefação



Fonte: Autor.

4.8 MATERIAIS E CUSTOS

Como efeito comparativo o protótipo desenvolvido teve um custo total em materiais (componentes) de R\$ 410,20. A discriminação dos valores dos componentes do sistema estão descritos na Tabela 10.

Tabela 10 – Discriminação dos valores dos componentes do protótipo.

Componente	Preço Médio (R\$)
Arduino Mega	89,90
ESP 8266-01	26,90
RTC DS3231	21,90
Módulo de cartão micro SD	8,90
4 DHT 22	128,00
2 DS18B20	29,80
16 Metros de cabos	65,00
Jumpers	19,90
Protoboard	19,90
TOTAL	410,20

Fonte: Autor.

5. CONCLUSÃO

Com o término deste trabalho, acredita-se que os objetivos propostos inicialmente foram cumpridos. O sistema desenvolvido é formado por duas partes, sendo a primeira parte a parte *online* via *Wi-Fi* que utiliza o ESP, fazendo conexão com dois *Websites* diferentes, sendo eles o *Google Planilhas* e o *ThingSpeak*. Para a melhor visualização dos dados, foi desenvolvido um *Website* através da plataforma do *WordPress*, com os gráficos realizados em tempo real utilizando o site *ThingSpeak*, em conjunto com um *link* que resulta em uma planilha *Google* para uma análise mais completa e elaborada dos dados obtidos. Com relação a parte *off-line*, o sistema é composto por um *datalogger* utilizando um cartão SD e um *Real Time Clock (RTC)*.

Todos os dados coletados foram salvos com sucesso tanto na parte *online* quanto na parte *off-line*. Grande parte dos dados possui uma excelente confiabilidade, e quando há algum tipo de interferência, pode ser facilmente percebida através da análise gráfica que é realizada automaticamente pela parte *online*.

Com relação ao estudo da casa virada para o norte, detectou-se com sucesso influência na temperatura interna dos cômodos da Casa Eficiente. Este estudo foi confirmado anteriormente através de um projeto realizado em 2015, quando ainda não havia moradores na residência e pôde através deste trabalho ser confirmado mais uma vez em 2018. Todavia, agora com a casa já possuindo moradores que interferem nas temperaturas dos cômodos diretamente ou indiretamente. Detectou-se também que há uma influência de outro projeto que se encontrava concluído, este projeto fez com que o quarto frontal acabasse por ser um dos mais quentes da casa.

Por fim, ressalta-se a importância da *IoT* neste trabalho, pois foi através dela que as análises foram possíveis de serem realizadas, com gráficos e dados em tempo real, facilitando muito este projeto. Também ressalta-se a importância da parte do *off-line* do *datalogger*, garantindo melhor segurança no armazenamento quando não havia uma conexão estável com a Internet. Para a facilitar a análise destes dados, os macros criados para isto funcionam com perfeição, trocando a tabulação e importando os dados dos blocos de notas criados. Acredita-se que tanto a implementação *online* quanto *off-line* ainda possuem muito o que evoluir tecnologicamente, mas mesmo com a utilização de componentes simples e considerados baratos, pôde-se obter um resultado considerado de boa qualidade neste trabalho.

REFERÊNCIAS

- ÁVILA. **10 exemplos de macros essenciais no Excel**. Disponível em <<https://blog.luz.vc/excel/10-exemplos-de-macros-essenciais-no-excel/>>. Acesso em 23 de Setembro de 2018.
- BAPTISTA. **SISTEMAS DE AQUISIÇÃO DE DADOS**. Relatório técnico. Escola Superior de Tecnologia de Viseu. Viseu, Portugal.
- CCM. **O protocolo TCP** Disponível em <<https://br.ccm.net/contents/284-o-protocolo-tcp>>. Acesso 23 de Setembro de 2018
- DEV MEDIA. Instalação do WampServer. Disponível em <<https://www.devmedia.com.br/instalacao-do-wampserver/25871>>. Acesso em 23 de Setembro de 2018
- DILLY, R.O; MENDES, L.F. **Aplicação em Tempo Real de Monitoramento de Umidade e Temperatura Utilizando Arduino**, 2015.
- DR.BIT, **ThingSpeak com Arduino e outros dispositivos**, Disponível em <<https://drbitblog.wordpress.com/2014/05/02/thingspeak-com-arduino-e-outros-dispositivos/>>. Acessado em 23 de Setembro de 2018
- EMBARCADOS, **Entradas e saídas do Arduino UNO** Disponível em <<http://www.embarcados.com.br/arduino-uno/>> Acessado em 22 de Setembro de 2018
- EMBARCADOS, **Arduino primeiros passos** Disponível em <<http://www.embarcados.com.br/arduino-primeiros-passos/>> Acessado em 22 de Setembro de 2018
- ENDEAVOR. **Tudo o que você precisa saber sobre internet das coisas**. 2015. Disponível em <<https://endeavor.org.br/tecnologia/internet-das-coisas>>. Acesso em 22 de Setembro de 2018
- EVANS, M., NOBLE, J. E HOCHENBAUM J. - **Arduino em Ação**, ed. Novatec 2013.
- HOSTMEDIA. **O que é WordPress? Tudo que você precisa saber sobre o assunto**. Disponível em <<https://www.hostmidia.com.br/blog/o-que-e-wordpress>>. Acesso em 23 de Setembro de 2018
- IBGE. **De 2005 para 2008, acesso à Internet aumenta 75,3%** <<https://censo2010.ibge.gov.br/noticias-censo.html?busca=1&id=1&idnoticia=1517&t=2005-2008-acesso-internet-aumenta-75-3&view=noticia>> Acesso em 21 de Setembro de 2018

MAGNUS, T. **O que é IoT (Internet das Coisas)? Futuro ou Presente?** Disponível em <<https://transformacaodigital.com/o-que-e-iot-internet-das-coisas/>>. Acesso em 23 de Setembro de 2018.

MARTINS, C. A. [**Introdução a aquisição de dados**].2013. 129 slides.

MAXIM INTEGRATED, **DS3231 Datasheet**, Electronic Publication, 2015

MCROBERTS, M. **Arduino Básico** ed. Novatec 2011

EIS, D. **O básico: O que é HTML?** Disponível em <<https://tableless.com.br/o-que-html-basico/>>. Acesso em 23 de Setembro de 2018.

NASCIMENTO, D. **Sistemas de aquisição de dados-Data logger**. Artigo científico. Acessado em <<http://www.ebah.com.br/content/ABAAAfnHAAJ/sistemas-aquisicao-dados-data-logger>>. Acesso em 07 de maio de 2018

PEREIRA, A. O que é Java? Disponível em <<https://www.tecmundo.com.br/programacao/2710-o-que-e-java-.htm>>. Acesso em 23 de Setembro de 2018.

SANTOS, J.C.P, *et al.*, **AVALIAÇÃO TÉRMICA DE CASA POPULAR EFICIENTE NA ZONA BIOCLIMÁTICA 2 BRASILEIRA NO PERÍODO DE VERÃO**. 2015

SAS. **Big Data**. Disponível em <https://www.sas.com/pt_br/insights/big-data/what-is-big-data.html>. Acesso em 22 de Setembro de 2018

SILVEIRA. C. B. **O Que é Indústria 4.0 e Como Ela Vai Impactar o Mundo**. 2017.

Disponível em <<https://www.citisystems.com.br/industria-4-0/>>. Acesso em 22 de Setembro de 2018.