



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SÃO PAULO
Campus Catanduva

PEDRO HENRIQUE SECCHI

PROTÓTIPO DE UM DISPOSITIVO PARA TELEMETRIA DE TEMPERATURA CORPORAL

CATANDUVA - SP

2019

PEDRO HENRIQUE SECCHI

PROTÓTIPO DE UM DISPOSITIVO PARA TELEMETRIA DE TEMPERATURA CORPORAL

Parte manuscrita do Projeto de pesquisa do aluno PEDRO HENRIQUE SECCHI, apresentado ao Instituto Federal de Ciência e Tecnologia de São Paulo Campus Catanduva, a ser utilizado com diretriz para a manufatura do Trabalho de Conclusão de Curso (TCC).

Orientador: Jair de Martin Jr

CATANDUVA - SP
2019

PEDRO HENRIQUE SECCHI

PROTÓTIPO DE UM DISPOSITIVO PARA TELEMETRIA DE TEMPERATURA CORPORAL

Parte manuscrita do Projeto de pesquisa do aluno PEDRO HENRIQUE SECCHI, apresentado ao Instituto Federal de Ciência e Tecnologia de São Paulo Campus Catanduva, a ser utilizado com diretriz para a manufatura do Trabalho de Conclusão de Curso (TCC).

Aprovada em (dia), de (mês) de (ano).

Jair de Martin Jr
Orientador

Professor
Convidado 1

Professor
Convidado 2

CATANDUVA - SP
2019

Resumo

Aproximadamente 25% das consultas nas emergências são devido a febre e em atendimentos pediátricos ela é um dos sintomas mais comuns nas crianças, porém em sua grande maioria, trata-se de um sintoma decorrente de uma doença benigna e auto limitante, que são aquelas que após o seu aparecimento duram entorno de três dias, sendo assim são bem poucos os casos de sintomas febris, decorrente de infecções bacterianas graves, que caso tenha atraso no diagnóstico pode causar sequelas ou até mesmo a morte; já os pais/-cuidadores, pela sensação de insegurança, preocupação e associação da febre sempre com doenças graves, acabam sofrendo de "febre fobia". O presente trabalho tem como objetivo o desenvolvimento de um protótipo capaz de medir a temperatura da criança de forma precisa e constante; buscando, gerar mais tranquilidade e conforto para os responsáveis, uma vez que o projeto conta com uma um conjunto de orientações baseado no artigo Critérios de alerta na criança com febre: texto para pais de Manuel Salgado (2014). Para isso será utilizado o microcontrolador ESP 01 que já conta com WiFi em seu chip e comunicação I2C, possibilitando a comunicação tanto com o sensor de temperatura, quanto com a aplicação instalada em um smartphone.

Palavras-chave: ESP01, Febre infantil, Termômetro.

Sumário

	Sumário	1
1	INTRODUÇÃO	3
1.1	CONTEXTUALIZAÇÃO	3
1.2	OBJETIVO	5
1.2.1	OBJETIVO GERAL	5
1.2.2	OBJETIVO ESPECÍFICO	5
1.3	JUSTIFICATIVA	5
2	FUNDAMENTOS TEÓRICOS	9
2.1	IoT(internet das coisas- Internet of Things)	9
2.1.1	Computação Ubíqua	9
2.2	Microcontroladores	11
2.2.1	Microcontrolador PIC	11
2.2.2	Plataforma Arduino	12
2.2.3	ESP8266	12
2.3	Sensores de Temperatura	13
2.3.1	RTDs	13
2.3.2	Termistores	14
2.3.3	Cls Sensores	15
2.3.4	Pares termoeletricos	15
3	MATERIAIS E MÉTODOS	17
3.1	ESP8266	17
3.2	Sensor TMP102	17
3.3	Interface	18
3.4	Arduino IDE	19
3.4.1	Sensor e Microcontrolador	19
3.4.2	Aplicativo	21
4	RESULTADOS	23
4.1	Circuito	23
4.2	Aplicativo	24
4.3	Aquisição de Dados	26
4.4	Programação do Microcontrolador	27
5	CONCLUSÃO	29

Referências	31
------------------------------	-----------

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

As queixas com relação a febre está entre as mais comuns em atendimentos pediátricos. Ela é vista com temor, por ser um sintoma inicial de doenças graves; apesar de que na maioria das vezes ser um sintoma de infecções virais agudas. Bricks (2006).

Estima-se que 25% das consultas nas emergências é devido a febre e em sua maioria, pode ter sua origem identificada logo na primeira avaliação médica, no entanto em aproximadamente 20% dos casos não é possível realizar este diagnostico logo no primeiro contato entre médico e o paciente, esta situação é conhecida como febre sem sinais localizatórios (FSSL).

Grande parte das crianças com FSSL, estão na fase inicial de alguma doença infecciosa benigna ou apresentam doenças infecciosas agudas auto limitantes. São bem poucas as infecções bacterianas graves (IBG), que neste caso, se houver atraso no diagnostico, há o risco de mortalidade Machado et al. (2009).

A associação do sintoma da febre com uma doença grave faz com que ela seja acompanhado de uma sensação de insegurança, ansiedade e preocupação por parte dos pais/-cuidadores, fazendo com estes procurem atendimento médico com urgência sem necessidade. Estes pais e cuidadores sofrem de "febre fobia", muitos tem em mente concepções antigas e sem fundamentos. SILVA et al. (2018)

Estima-se que hajam mais de 400 vírus causadores da febre, porém nem toda criança que contrai uma infecção viral, tem febre como sintoma. Um exemplo é a varicela, que em algumas crianças geram o sintoma da febre, enquanto em outras não, é por esse motivo que algumas crianças saudáveis terão mais febres do que outras. A maioria das doenças febris, são benignas e causados por vírus, e não necessitam de antibióticos Salgado (2014).

O corpo humano é ineficiente quanto a medição de temperatura, uma vez que se pode apenas concluir se algo está mais quente ou frio do que sua própria temperatura, que por sua vez pode variar dependendo da atividade exercida anteriormente. Por isso houve-se a necessidade da criação de um instrumento que utilizasse de outros métodos para aferir a temperatura. O primeiro registro histórico se dá em II a.C., onde um médico grego chamado de Galeno sugeriu duas escalas com quatro divisões, ambas partindo de um ponto neutro, o ponto mais alto desta escala se dá com a fervura da água e o mais baixo com a temperatura de congelamento da água e o neutro sendo a média. Por volta do ano de 1100 outro médico Harme de Berna, utilizou-se de uma escala semelhante a de Galeno, porém os "4 graus de calor"(como era nomeado o ponto onde se encontrava a maior temperatura da escala) tinha como referência a temperatura do equador e os "4

graus de frio" os polos terrestres; essa escala estabelecia a mistura ideal de drogas para ministrar.

Galileu Galilei em 1592, inventou o termoscópio, que utilizava a expansão do ar, para medir a temperatura. Apesar de ser o mais lembrado diversos outros como, Sanctorius Santori da Capodistria (1561-1636), Cornelius van Drebbell (1572-1634), Robert Fludd (1574-1637), Giambalista Porta (1538-1615), também são apontados como os idealizadores do termômetro.

Em 1606, Porta descreveu em seus registros detalhes do termoscópio, este consistia de um bulbo, com um longo tubo com sua abertura mergulhada em um líquido, que poderia ser água ou álcool e posteriormente vinho, expulsava-se um pouco de ar do tubo, fazendo com que o líquido de aferimento alcançasse um certo nível, a partir daí conforme o instrumento aquecia e esfriava, o nível do líquido subia ou descia, porém este instrumento sofria com a pressão atmosférica por ser aberto para o meio, além do líquido evaporar com o tempo. Pires et al. (2006)

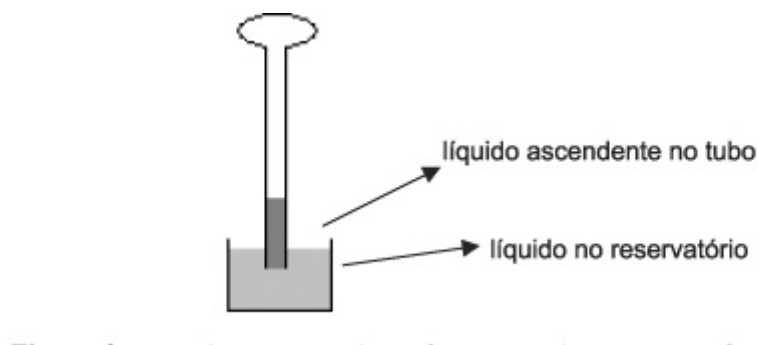


Figura 1 – Desenho esquemático de um termoscópio.

Partindo deste princípio as principais evoluções que trouxe ao termômetro que conhecemos se deram através de:

- Robert Boyle (1627-1691) que estabeleceu o ponto de fusão como ponto fixo;
- Joachin Dalence (1640-1707), em 1688 destacou a necessidade de dois pontos fixos para estabelecer uma escala, que inicialmente eram do ponto de fusão da água e da manteiga
- Renaldini (1615-1698) substituiu o ponto de fusão da manteiga pela ebulição da água, pois segundos estudos feitos por Robert Hooke (1635-1703) os valores de fusão e ebulição da água se mantinham constantes;

Pires et al. (2006)

Em 1638, Sanctorius foi quem deu início a pesquisa sobre a temperatura corporal, foi ele quem percebeu as variações de temperatura, foi a partir disso que ele deu início a utilização de instrumentos para verificar tais variações.

Wunderlich em 1851, deu início a utilização do termômetro de mercúrio, introduzindo-o axila para realizar a medição, além de definir o valor de 38 °C ou superior, como estado febril.

A cada dia, os antigos termômetros de vidro-mercúrio, que causam danos quando quebram como lacerações e/ou intoxicação estão sendo substituídos por instrumentos digitais, que são mais seguros rápidos e de fácil manuseio. Simões and De Martino (2007)

1.2 OBJETIVO

1.2.1 OBJETIVO GERAL

Este projeto tem como propósito o monitoramento da temperatura corporal de crianças e bebês, principalmente durante o sono e em períodos que houver suspeita de febre, auxiliando também na classificação com relação da gravidade deste sintoma, fazendo com que os pais tenham uma melhor orientação evitando assim desgastes físicos e/ou emocionais.

Para isso o equipamento de telemetria deve ser leve e relativamente pequeno, e enviará as informações via internet para um software que contará com um questionário para a classificação dos possíveis agentes causadores da febre, para que haja uma complemento no momento do pré-diagnóstico.

1.2.2 OBJETIVO ESPECÍFICO

1. Contribuir no auxílio do diagnóstico de alguns tipos de febres;
2. Utilizar de um microcontrolador que facilite a comunicação com a internet;
3. Mostrar a temperatura da criança no decorrer da utilização do equipamento;
4. Construir uma circuito unindo o microcontrolador, o sensor e a alimentação;
5. Utilizar internet para a comunicação do sensor com a estação que fará a interface homem máquina;
6. Desenvolver um software para monitoramento e algumas orientações para os pais e responsáveis.

1.3 JUSTIFICATIVA

Entre as maiores causa de procura pediátrica, está a febre, segundo o que Magni retrata em seu artigo, das crianças com menos de 36 meses que são levadas ao médico pelos responsáveis, aproximadamente 2/3 são por causa deste sintoma, que nada mais é do que

o aumento da temperatura do corpo humano em resposta a várias doenças, estas podendo ser ou não infecciosas.

Murahovschi (2003) também relata que a febre, trata-se do aumento de temperatura acima dos considerados normais, porém estabelecer um valor normal não é algo tão fácil, pois esta varia de valor dentro de alguns limites, dependendo de alguns fatores como:

- Horário: a temperatura corporal tende a ser mais alta no final da tarde (17 horas) e atingir os valores mais baixos durante a madrugada (3 horas);
- Idade: lactantes com menos de um ano tem temperatura normal superior;
- Gênero: no caso do sexo feminino as temperaturas são comumente mais altas do que no sexo masculino, além disso, sofre alterações durante o ciclo menstrual;
- Ambiente: fazer muitas atividades em locais pouco ventilados, quentes tende a aumentar a temperatura.

Murahovschi (2003) ainda indica alguns parâmetros para as medições de temperatura corporal: a temperatura axilar normal varia $36,5^{\circ}\text{C}$ à $37,2^{\circ}\text{C}$ dependendo da hora do dia sendo a temperatura bucal aproximadamente $0,5^{\circ}\text{C}$ a mais do que a axilar e a retal varia entre $0,8^{\circ}\text{C}$ e $1,0^{\circ}\text{C}$ a mais do que a axilar, ou seja uma temperatura maior que $37,2^{\circ}\text{C}$ pode-se considerar um estado febril.

Como Casanova et al. (2014) descreve, a febre trata-se de um importante sinal fisiológico do organismo a um agente, este podendo ser infeccioso ou não, Além de servir como um sinal, o mesmo também dificulta a reprodução microbiana (em caso de infecção).

Complementando a ideia anterior Salgado (2014) afirma ainda a existência de 400 vírus causadores de febre infantil.

Salgado (2014) ainda define dois tipos de doenças febris:

1. Doenças febris benignas provocadas por vírus: são aquelas responsáveis pela maioria dos casos de febre e não necessita do uso de antibióticos;
2. Doenças febris graves (as meningites e/ou as sépsis): estas podem colocar em risco a vida e/ou provocar lesões permanentes, e felizmente são raras.

Salgado (2014) ainda retrata através de gráficos a seguintes situações:

A figura 2 representa 1.456 casos de doenças autolimitada, ou seja, aquelas que não necessitam de medicação, este dados foram colhidos em um período de 12 anos (1996-2008) em uma consulta privada informatizada de ambulatório não hospitalar. O objetivo deste, era computar a duração da febre neste caso. Em 51 % dos casos a doença não ultrapassa quatro dias.

Na figura 3 mostras a duração da febre no ato da internação no Hospital Pediátrico de Coimbra em um período de 21 anos, dos 419 casos de febres graves, 65% já foram

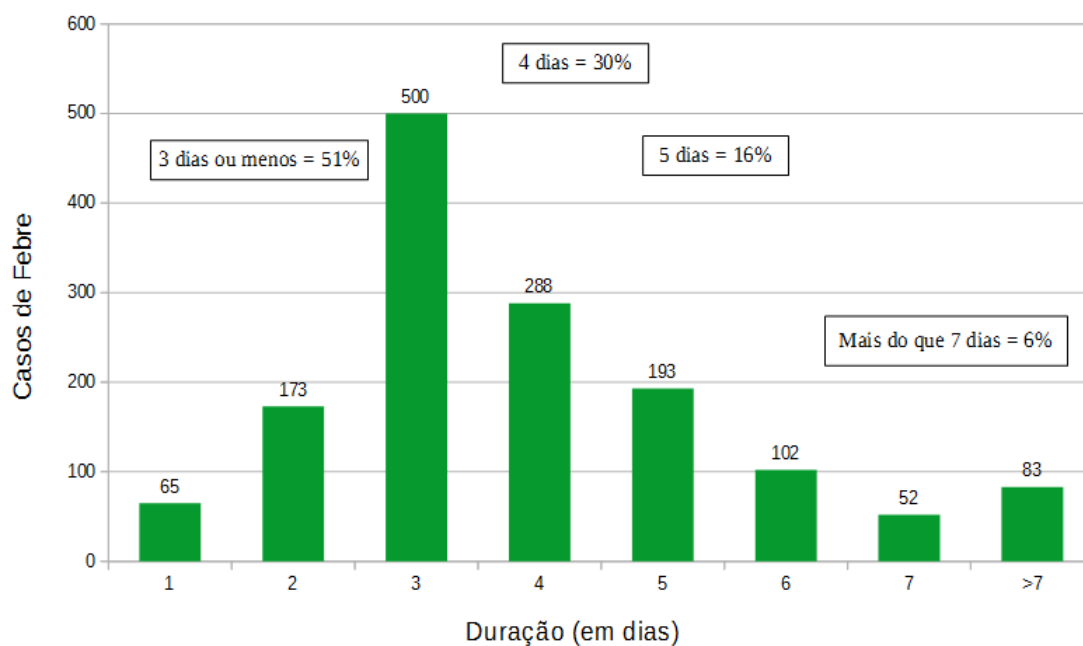


Figura 2 – Duração, em dias, de 1.456 episódios de doenças febris benignas autolimitadas.

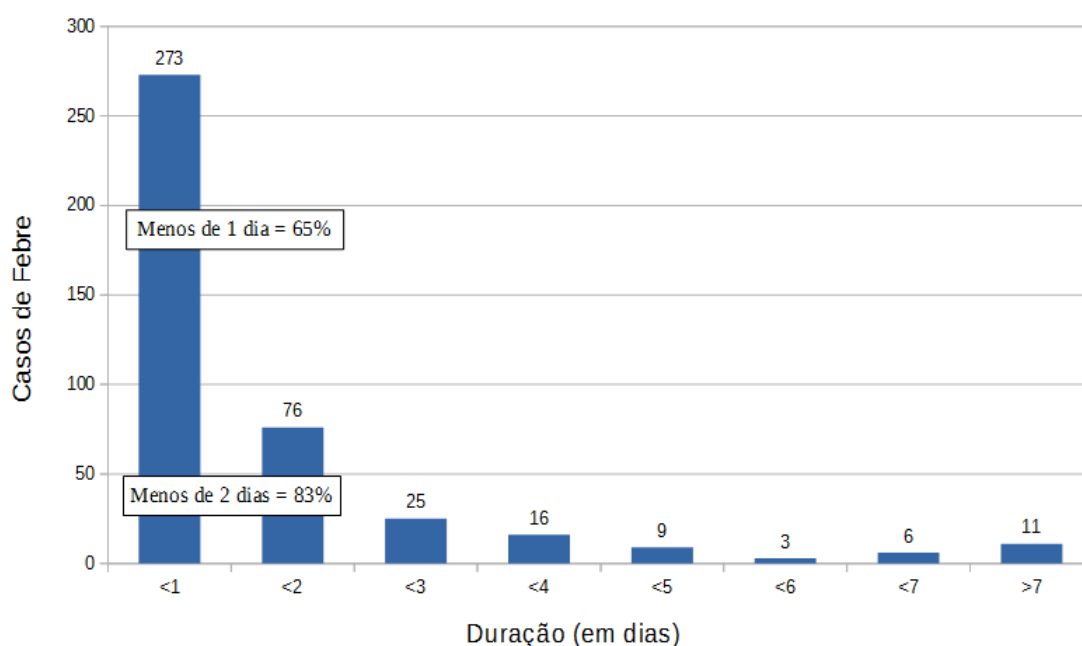


Figura 3 – Duração da febre (em dias) no momento do internamento em 419 doenças graves internadas no HPC.

internados em menos de 24 horas e com duração de menos de 48 horas, 83% dos pacientes já estavam internados.

Ou seja, Salgado (2014) refuta a teoria de só ir ao médico após os três dias de febre, algo muito disseminado entre os pais, pois:

- Para as doenças ligeiras é demasiado cedo;

- Para as doenças graves é demasiado tarde.

2 FUNDAMENTOS TEÓRICOS

Nesse capítulo será descrito os conceitos utilizados no projeto.

2.1 IoT(internet das coisas- Internet of Things)

Segundo de Oliveira (2017) descreve em seu livro, a internet das coisas, é um conceito de integração de vários dispositivos através da internet, não só promovendo a comunicação pura e simples, mas também fazendo o processamento das informações que trafegam de um dispositivo para outro de maneira inteligente. Apesar de parecer um conceito novo, ele já era almejado há vinte anos, quando a internet começava a se popularizar, porém somente agora, com os baixos custos dos componentes, a IOT vem ganhando força.

Para Ferreira (2014) a IOT, é a ideia de que cada objeto deve ter apenas um endereço e um endereço pertencer a apenas um objeto, com a comunicação se dando através da internet, esta tecnologia se ampliou ainda mais, permitindo não somente a comunicação entre as pessoas mais também de pessoas com objeto e de objetos com objetos.

A comunicação entre objetos (M2M- machine to machine) proporcionará a possibilidade de execução de serviços complexos, zerando as intervenções humanas, permitindo que estes processos ocorram em um segundo plano, fora do foco do homem, tornando-os perversivos.

Uma segunda característica apontada por Ferreira (2014) é a mobilidade, gerada pelo uso da internet como meio de comunicação. Pelo fato da internet ter um alcance global, esta comunicação permite que o objeto se conecte com ela em qualquer lugar do mundo, podendo inclusive trocar sua geolocalização sem perder sua conectividade.

A fusão destas duas características, persividade e mobilidade, geram o conceito de ubiquidade criado por Mark Weiser, na década de 90

2.1.1 Computação Ubíqua

Criado em 1991 por Weiser, o conceito de computação ubíqua, é fundada por 3 pilares principais:

1. Alto grau de mobilidade;
2. Transparência para o usuário final;
3. Reação ao contexto.

Computação móvel é aquela em que permite a troca de geolocalização e até mesmo a troca de máquina, sem que esta altere suas funcionalidades e sua comunicação com a rede.

Já a computação pervasiva, traz a transparência e a facilidade no uso, além da capacidade de interpretar o meio onde ele está inserido.

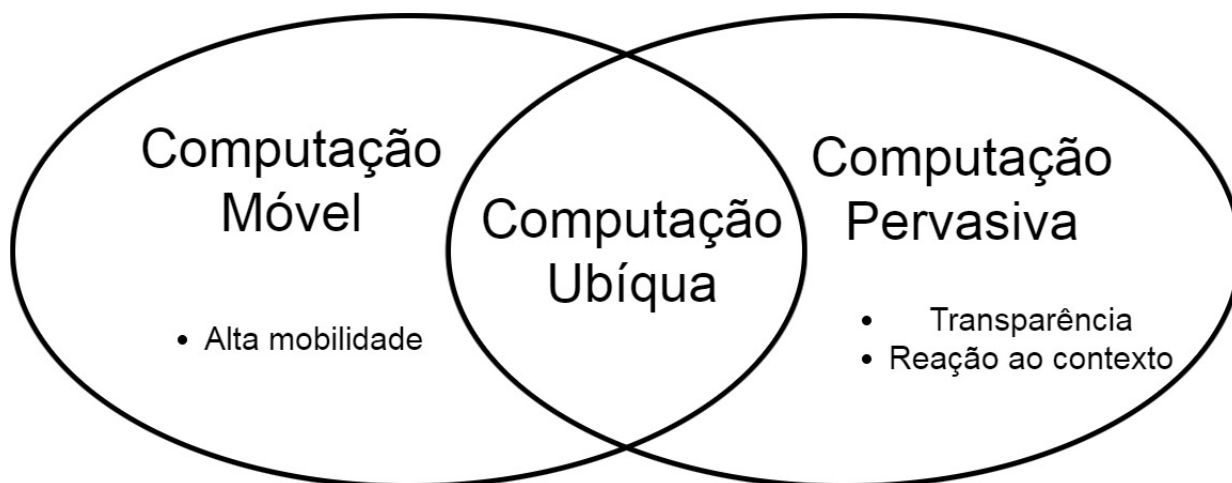


Figura 4 – Relação entre Computação Ubíqua, Pervasiva e Móvel.

Como demonstrado na figura 4 com a união destes elementos temos a ubiquidade.

Em seu artigo de Araujo (2003) relata os três princípios da computação ubíqua, sendo elas:

1. **Conectividade:** é o fato das aplicações se moverem juntamente como seu usuário por diversos e distintas redes, sem que seja necessário a intervenção do mesmo. Porém isto será possível apenas utilizando-se de aplicações com padrões comuns entre si.
2. **Diversidade:** atualmente o mais comum é que dispositivos desempenhem diversas funções, como nos casos dos PCs e dos smartphones, porém dispositivos ubíquos apresentam um propósito específico, apesar da possibilidade de haver mais de uma funcionalidade por dispositivos alguns são melhores para desempenham papéis do que outros. Outro aspecto é de como fazer o gerenciamento destes dispositivos levando em conta a capacidade de cada um, com diferentes limitações o que torna desenvolver uma aplicação comum há eles.
3. **Descentralização:** A distribuição entre vários dispositivos é fundamental na computação ubíqua, ou seja, vários dispositivos com diferentes funções e desenvolvendo diferentes tarefas, porém trabalhando juntos, para isso conta com uma rede dinâmica onde os equipamentos se comunicam entre si e com um servidor caracterizando um sistema distribuído. Na computação ubíqua as responsabilidades são distribuídas entre vários.

2.2 Microcontroladores

de Oliveira (2017) relata que dos primórdios do que conhecemos hoje como microcontroladores se deu na verdade no início da década de 70 com a Intel e seus microprocessadores de baixo custo da época, além de outras empresas, em especial com a Texas Instruments com seu microcontrolador TMS 1000. Antes as funcionalidades dos microcontroladores se resumiam a suas interfaces de entrada e saída, mas com o tempo agregaram mais características, como memória RAM e EPROM, circuito de clock e interfaces de comunicação.

Juntando toda essa gama de funcionalidades em apenas um componente o torna muito barato e acessível e a quantidade ínfima de energia para o seu funcionamento, justifica o seu uso em larga escala.

Para Martins (2005), o fato dos microcontroladores desempenharem uma infinidade de funções, que antes necessitariam um circuito complexo, faz com que aprender a programá-los se assemelhe a aprender a resumir circuitos em um único componente. Ele ainda nos dá a ideia de que microcontroladores são pastilhas contendo processador, memória e pinos de entrada e saída, controlados por meio de programação e de uma referência interna ou externa.

2.2.1 Microcontrolador PIC

A Microchip é a responsável pelos microcontroladores da família PIC, que graças a sua linguagem Assembly mais simplista do que os demais fabricantes e o MPLAB IDE (software utilizado para a criação de programas), se tornou muito popular.

O PIC utiliza-se de uma estrutura RISC (Reduced Instruction Set Computer) que baseasse em barramentos diferentes para dados e programa. Martins (2005)

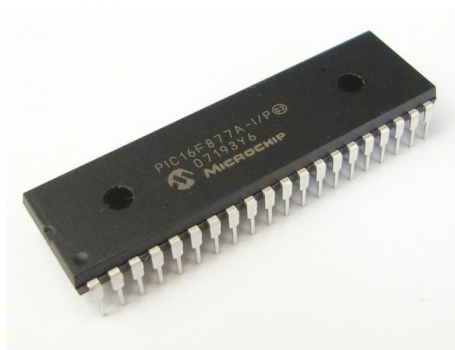


Figura 5 – Microcontrolador PIC16F877A.

2.2.2 Plataforma Arduino

Criado em 2005 o Arduino Project se popularizou rapidamente Itália a fora; estima-se que mais de 150 mil placas oficiais e mais 500 mil não oficiais e algumas de suas variantes tenham sido comercializados até o ano de 2011. Dentre os grandes motivos para sua popularização tão rápida está a sua fácil utilização por pessoas que não são da área técnica, além da sua curva de aprendizagem rápida. A plataforma Arduino consiste de duas vertentes:

- Hardware: Que utiliza um Microcontrolador ATMELE AVR, além de uma placa simples, que pode ser replicada, por se tratar de um Hardware livre.
- Software: Consiste de uma linguagem semelhante ao C (que é muito difundida) e uma IDE própria.

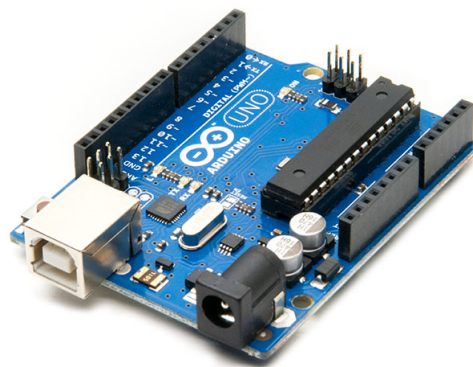


Figura 6 – Placa de desenvolvimento Arduino Uno.

2.2.3 ESP8266

Produzido pela Espressif Systems, o ESP8266 tem como diferencial comunicação WiFi embutida em seu chip, por isso ele ainda é largamente utilizado como módulo para outros microcontroladores, como o Arduino, outra grande vantagem é o custo que varia de 20 à 50 reais.

Por estar no mercado desde 2014, isso faz com que os trabalhos e documentações sobre eles ainda sejam escassos.

Existem diversos tipos de ESPs, o que divergem entre eles é a quantia de pinos disponíveis, bem com o espaçamento entre eles, além da memória disponível. Oliveira (2017)

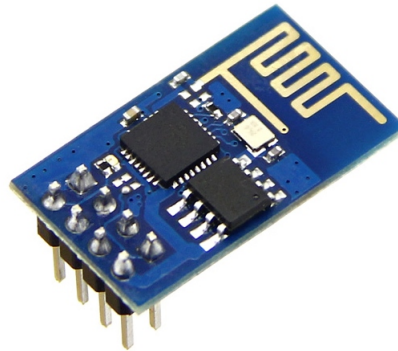


Figura 7 – Placa de desenvolvimento ESP8266 01.

2.3 Sensores de Temperatura

Segundo Braga (2018) descreve em seu artigo, são diversos fatores que interferem na escolha de um sensor de temperatura e estes, se mal avaliados, podem resultar em dados inesperados e consequentemente em um projeto desastroso. Ele afirma ainda que a grandeza temperatura não é algo tão simples de ser medido quanto parece e que isso se dá por ter diversos fatores que devem ser levados em conta como: ruídos, taxa de repetição da medição, o tempo correto necessário para as medições, ou seja, ao escolher o sensor o projetista deve tomar alguns cuidados.

Nesse artigo Braga (2018) ainda descreve quatro tipos de sensores:

2.3.1 RTDs

Resistance Temperature Detectors ou simplesmente RTDs são sensores que utilizam a variação da resistência de acordo com a temperatura, sendo os mais baratos feitos de níquel e ligas de níquel, enquanto os mais caros e com de maior qualidade ,são feitos de platina.

A grande desvantagem deste tipo de sensor é a necessidade da passagem de corrente pelo circuito o que pode causar auto aquecimento, interferindo nas medições, outro problema é o fato do fio sensor ser muito curto, o que acarreta em uma resistência baixíssima que traz dificuldades no momento da elaboração do circuito, além da própria resistência cabos de conexão que passam a influenciar nas medições, ou seja, utilizarmos o circuito, semelhante a figura 8 a própria resistência do cabo será somado a medida do sensor, dando um valor incorreto.

Para medir efetivamente a resistência dos terminais do sensor, deve-se utilizar uma interface contendo quatro cabos, fazendo com que o valor não seja afetado pela resistência do fio, conforme demonstra o circuito da 9.

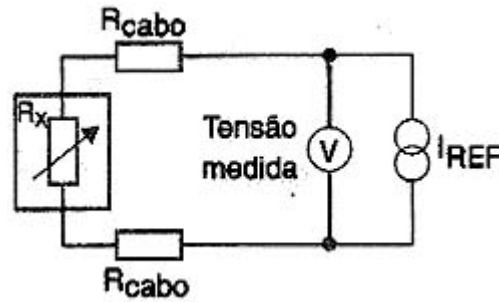


Figura 8 – Circuito simples de ligação do sensor RTD.

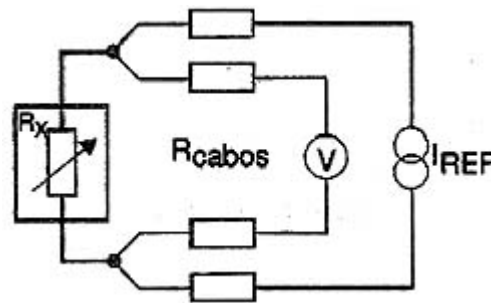


Figura 9 – Circuito com quatro cabos para a ligação do sensor RTD.

Há ainda uma terceira forma de se fazer a montagem deste circuito para evitar a influência das resistências dos cabos de conexão, utilizando apenas três deles como indica a figura 10.

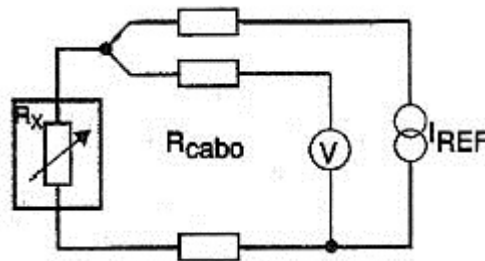


Figura 10 – Circuito com três cabos para a ligação do sensor RTD.

2.3.2 Termistores

Assim como os RTDs, termistores são sensores que utilizam a resistência para medir a temperatura, porém sua grande diferença está na material, que no caso dos termistores se trata de materiais cerâmicos semicondutores que apresentam uma resistência relativamente mais alta. Além disso, por terem um volume muito pequeno, a chance de causar divergência no material que se está medindo é menor. Porém, como os RTDs a alta corrente que passa pelo sensor e pelo fato de ser pequeno, pode haver divergência, causado pelo auto aquecimento. Outra desvantagem se dá pelo fato da resistência não variar na mesma proporção que a temperatura necessitando assim, de um algoritmo de correção.

2.3.3 CIs Sensores

O grande destaque dos CIs é que são projetados para emitir uma resposta linear com relação a temperatura, além dessas resposta serem emitidas em um sinal intenso, descartando a necessidade de outros circuitos de condicionadores de sinal para a medição. Em contrapartida a quantidade de tipos de componentes é pequeno, assim como os valores de medição também são muito restritos, Outra desvantagem é a necessidade de uma alimentação externa o que pode causar um auto aquecimento.

2.3.4 Pares termoelétricos

As características como robustez e a possibilidade de medir valores altos de temperatura fazem com que o pares termoelétricos ou termopar, tem um grande destaque frente aos demais. Ele é constituído de dois metais diferentes que constituem duas junções submetidas à temperaturas diferentes, gerando uma tensão proporcional a diferença das temperaturas, sendo assim, não necessitam de uma fonte de alimentação para o seu funcionamento.

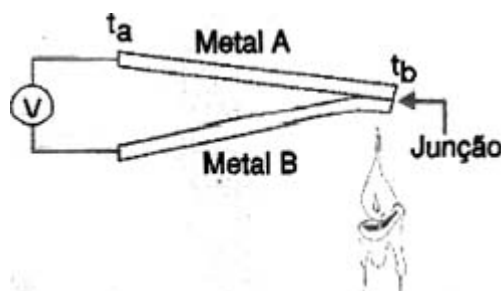


Figura 11 – Sensor de pares termoelétricos.

Como este sensor mede apenas uma diferença de temperatura, se faz necessário conhecermos uma das grandezas, seja a temperatura na chamada junta fria ou a tensão nesta mesma junta, fazendo com que possa se chegar no valor medido na junta quente. Para isso é necessário um circuito para chegar a estes valores outro para amplificar o valor de tensão na junta fria, que assim possa ser feito a medição com maior facilidade. Podemos imaginar assim, que qualquer problema decorrente destes dois circuitos pode gerar um erro na medição deste sensor.

3 MATERIAIS E MÉTODOS

Neste capítulo será descrito cada componentes utilizados no projeto.

3.1 ESP8266

Levando em consideração o fato da família ESP8266 conter uma comunicação com a rede wifi e sendo o ESP 01 um dos menores e um dos mais acessíveis, além de utilizar apenas 3,3V faz com que o ele seja ideal para este protótipo.

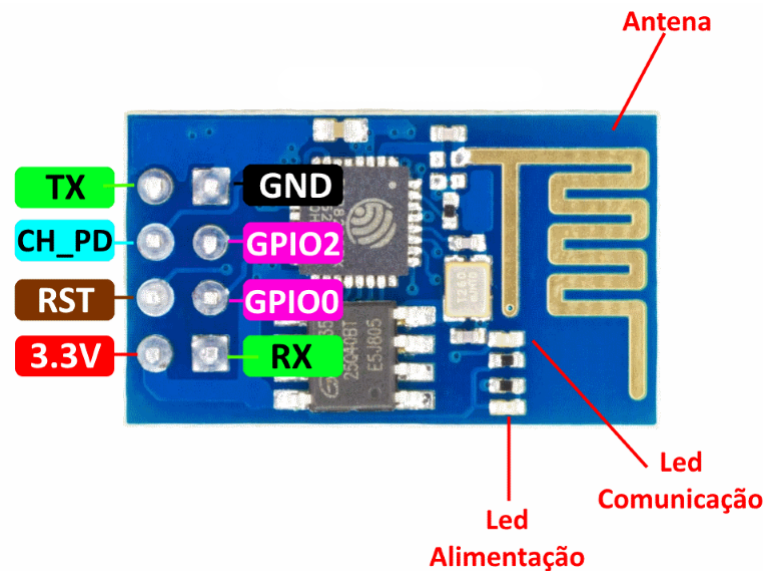


Figura 12 – Pinagem do ESP8266 01.

Apesar de ter poucos pinos GPIO, o ESP 01 tem interface I2C que acaba compensando este fator.

3.2 Sensor TMP102

Dentre as principais características encontradas no TMP 102 segundo o data sheet fornecido pela Texas Instruments estão:

- Medição de temperatura entre -40°C e 125°C ;
- Precisão de $\pm 40^{\circ}\text{C}$ sem a necessidade de condicionamento de sinal nem calibrações;
- Corrente de Operação de $10\ \mu\text{A}$;
- Tensão necessária varia de 1,4 à 3,6 V

- Um conversor analógico digital de 12 bits que possibilita uma resolução de até $0,0625^{\circ}\text{C}$
- Interface de comunicação I2C.

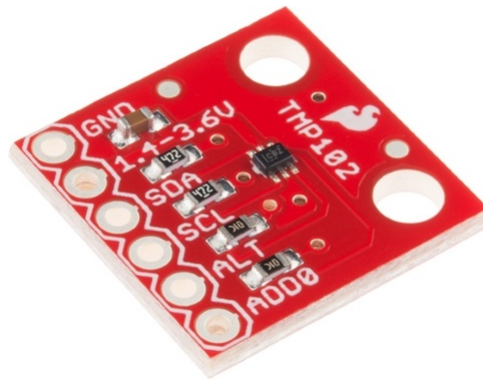


Figura 13 – Sensor TMP102

3.3 Interface

Para que seja possível verificar a temperatura será desenvolvido um aplicativo para smartphone, para isso levou-se em consideração a praticidade, se comparadas com outras possibilidade de IHM.

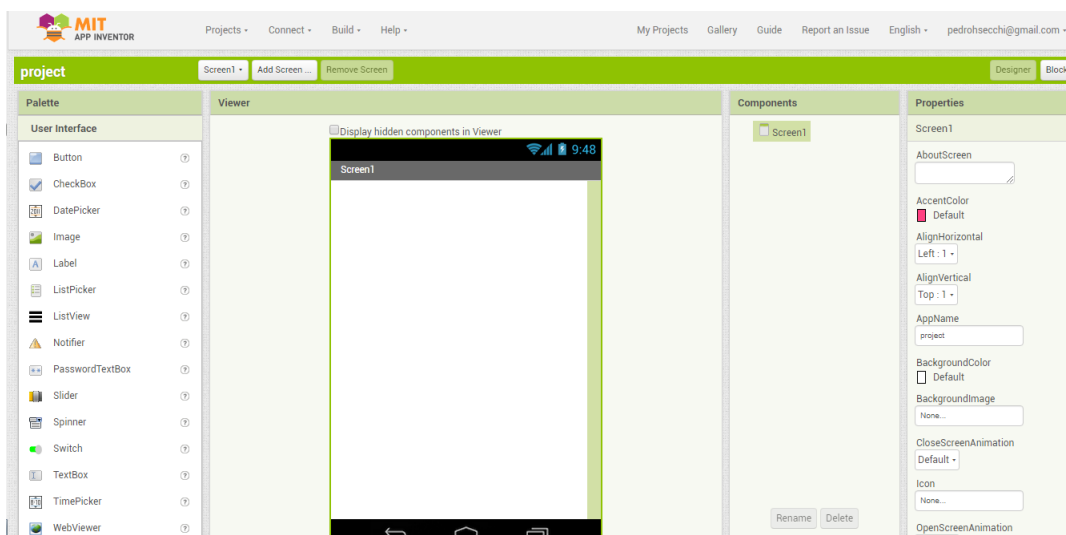


Figura 14 – Interface do APP Inventor

Para seu desenvolvimento será utilizado o ambiente de desenvolvimento APP Inventor, pois:

- Conta com uma interface intuitiva.

- É relativamente fácil de ser programado(se comparado com a forma clássica de programação).
- Compatível com a grande maioria de aparelhos que utilizam sistema Android
-

3.4 Arduino IDE

Para programação do microcontrolador Esp 01 será utilizado o Arduino IDE, que apesar de ser originalmente ter sido pensado para programação apenas de placas arduinos, a partir do uso de um plugin, também está sendo amplamente utilizado para a programação da família de microcontroladores ESP.

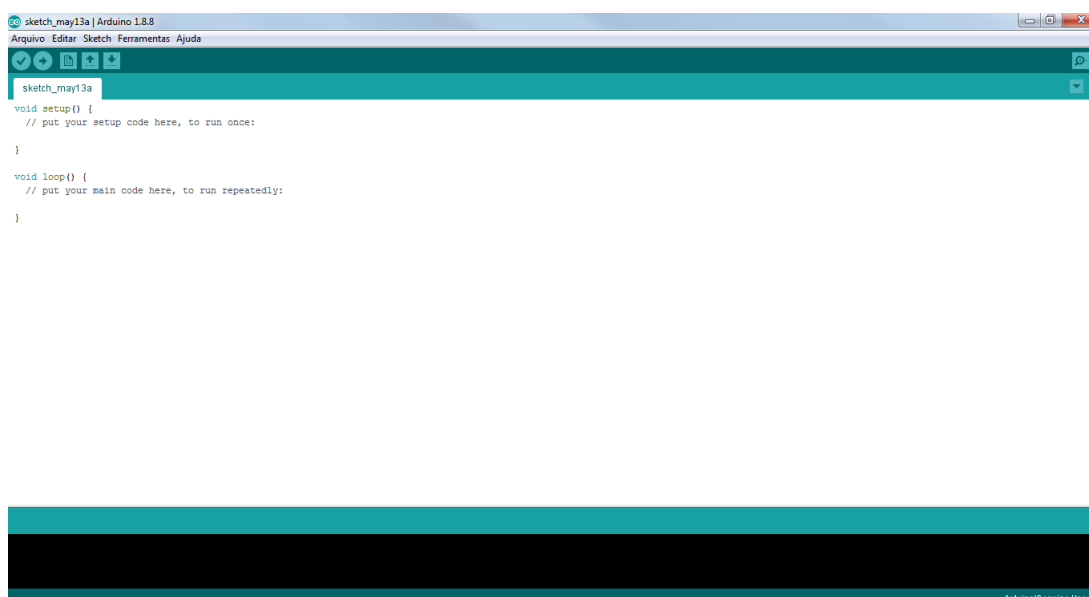


Figura 15 – Arduino IDE

O projeto contará com duas partes:

1. Sensor e Microcontrolador: que farão a medição da temperatura
2. Aplicativo: que será responsável pela IHM

3.4.1 Sensor e Microcontrolador

A figura 16 apresenta o fluxograma adotado para o microcontrolador.

Ao ser iniciado, o microcontrolador, verifica a conexão caso não esteja conectado, ele tenta estabelecer uma conexão até que consiga.

A partir conexão estabelecida, é requisitado pelo microcontrolador via comunicação I2C ao sensor o valor de temperatura, este envia ao ESP o valor dividido em dois bytes.

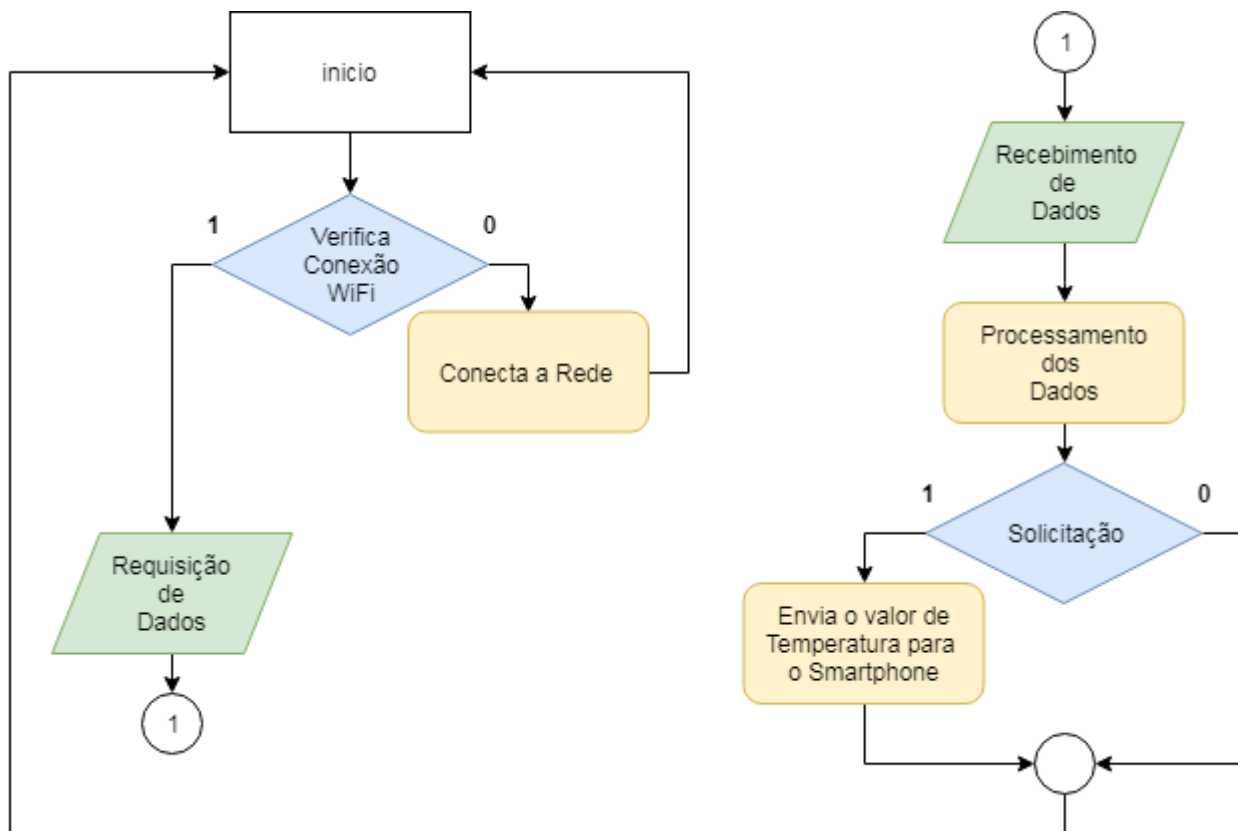


Figura 16 – Fluxograma do Esp01

Para processar deve-se seguir a equação descrita no datasheet do TMP102:

$$\text{MSB} \times (0.0625 \text{ }^{\circ}\text{C}/\text{LSB}) = \text{TMP}$$

Onde:

MSB é o valor dos doze bits mais significativos;

LSB é o valor dos quatro bits menos significativos;

TMP é o valor de temperaturas em Celsius.

Para chegar a este resultado é necessário alguns cálculos utilizando os dois bytes enviados pelo TMP102.

Primeiramente é necessário transformar os dois bytes, que originalmente estão em decimal, para binários e colocar cada bit em uma célula de um vetor. Após isto será necessário separar os dois bytes em um numero contendo os doze bits mais significativos e outro com os quatro bits menos significativos.

Antes de fazer os cálculos é necessário transformar estes dois números de binário para decimais, a partir destes valores é possível realizar o calculo.

Para finalizar loop, o microcontrolador verifica se há uma solicitação do valor de temperatura, em caso de positivo ele envia o valor para o aplicativo, do contrário ele retorna ao início do loop.

3.4.2 Aplicativo

O aplicativo solicita em um certo intervalo de tempo o valor de temperatura ao microcontrolador via rede WiFi, para isso no entanto será necessário que ambos estejam na mesma rede, para que haja comunicação.

Além de ser a interface do usuário, o mesmo também conta com algumas orientações apresentadas no artigo Critérios de alerta na criança com febre: texto para pais do autor Salgado (2014), para que assim seja possível trazer para os pais e responsáveis algum tipo de suporte emocional.

4 Resultados

4.1 Circuito

Para a integração de todos os componentes e módulos que compõe o projeto, foi elaborado um esquema elétrico demonstrado na figura 17

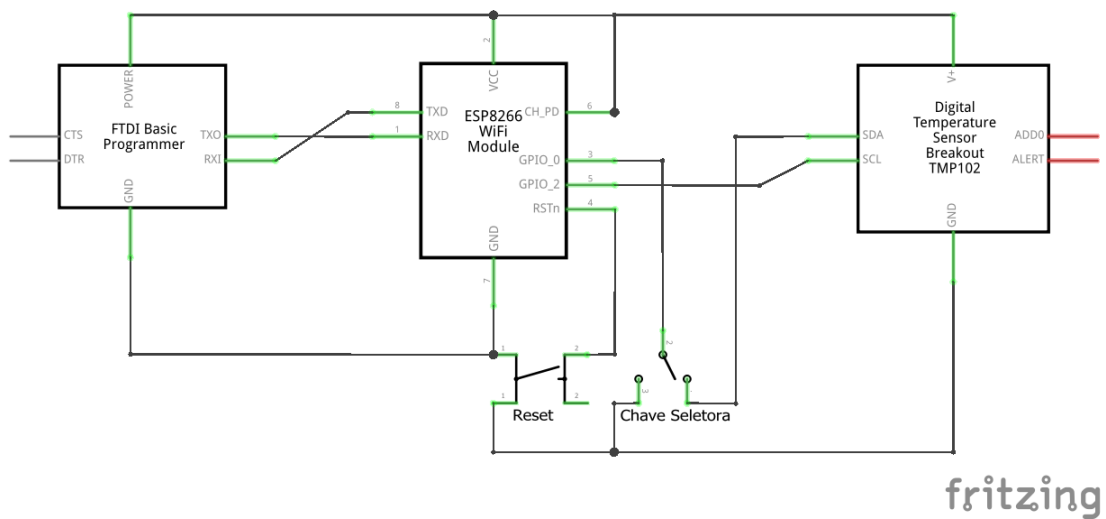


Figura 17 – Esquema elétrico

O circuito possui dois modos de funcionamento escolhido através da chave de seleção, uma vez que o pino GPIO 0 é conectado ao pino SDA do TMP102 ele está em modo de operação, quando este pino é aterrado o circuito encontra-se em modo de gravação, há também um botão, cuja a função é aterrar o pino Reset do microcontrolador, fazendo com que o mesmo reinicie e por fim, o protótipo utiliza uma placa FTDI FT232RL, semelhante a da figura 18 que nada mais é do que uma conversor USB- Serial, quando o circuito está em modo de operação, ela é apenas utilizada para energizar o mesmo e quando está em modo de gravação ela também faz a parte de conversão.



Figura 18 – Placa FTDI FT232RL

A figura 19 apresenta uma montagem em uma protoboard semelhante à utilizada para a realização de teste.

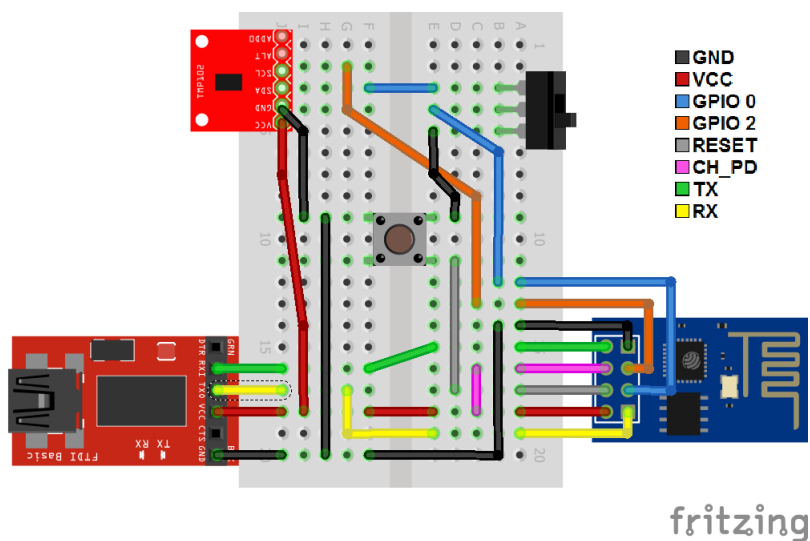


Figura 19 – Circuito

Utilizando o esquema foi elaborado uma PCB (printed circuit board - placa de circuito impresso), o que torna o projeto mais robusto e diminui consideravelmente os problemas decorrentes de alguns maus contatos devido a montagem na protobord. A imagem 20 mostra o desenho feito para a confecção da PCB.

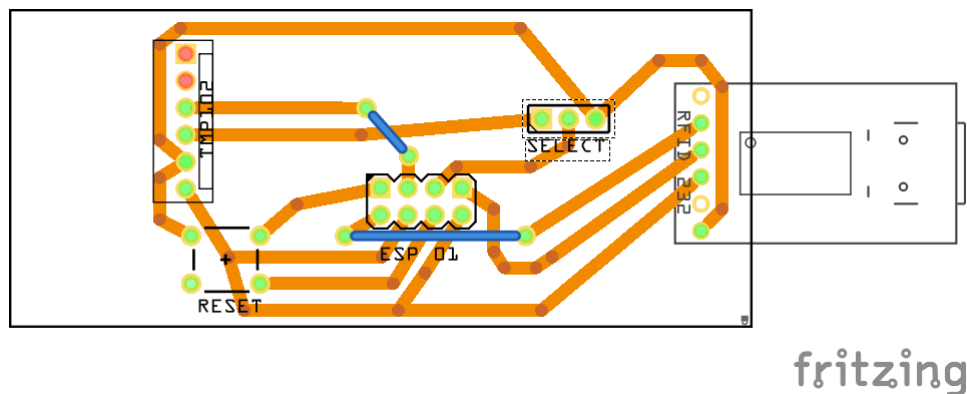


Figura 20 – Desenho utilizado para confecção da PCB

4.2 Aplicativo

Para realizar a consulta do valor de temperatura foi criado um aplicativo para smartphone, utilizando o ambiente App Inventor. Afim de criar uma interface intuitiva e simplificada chegou-se ao resultado mostrado na figura 21

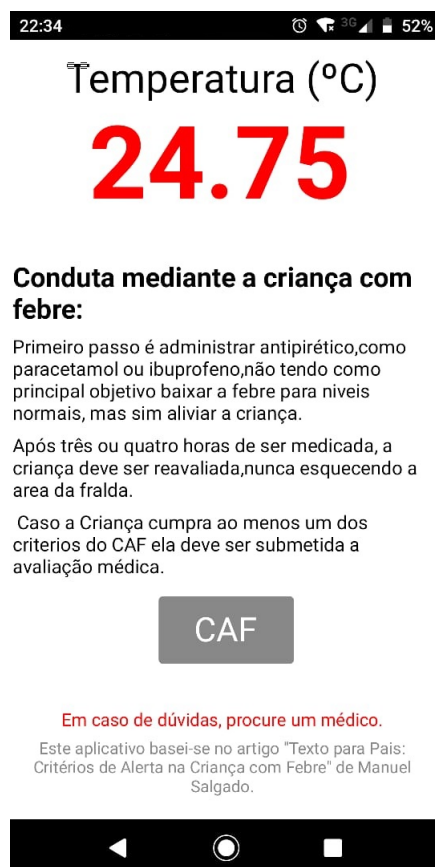


Figura 21 – Tela inicial do Aplicativo

Para o projeto é necessário que constantemente o aplicativo solicite o valor, para isso será utilizado o código conforme demonstra a figura 22

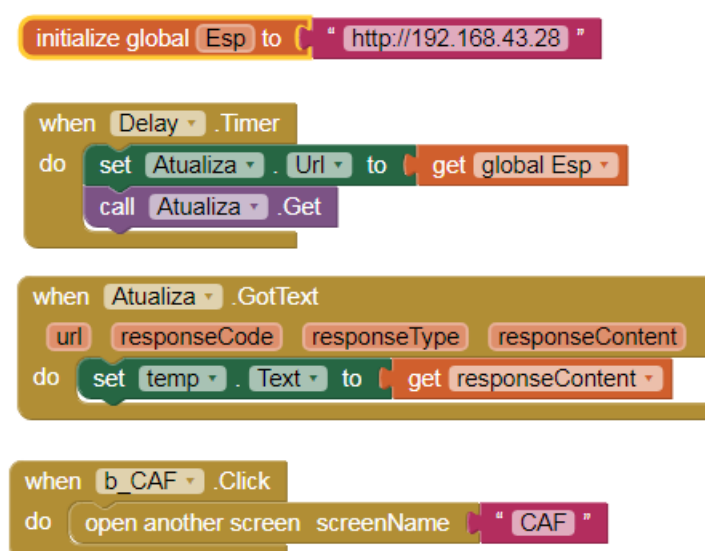


Figura 22 – Blocos de programação do AppInventor

No primeiro conjunto de blocos atribuiremos o valor do endereço IP do Esp 01 à variável global "Esp", o que facilita caso seja necessário alterar este endereço.

O segundo conjunto é controlado por um delay, ao ser acionado o mesmo envia uma solicitação ao microcontrolador, após isso ele chama o próximo conjunto, no qual ele recebe o valor de temperatura.

O último conjunto dos blocos é responsável por abrir a tela onde está o CAF (Critérios de Alerta da Febre).

4.3 Aquisição de Dados

Foi realizado um comparativo entre o protótipo e um termômetro de mercúrio, para isso adotou-se alguns parâmetros para diminuir algumas variáveis:

1. A aquisição do valor de temperatura foi realizada em de quatro ambientes distintos, afim de atingir valores diferentes entre os locais;
2. As medições foram realizadas em lugares fechados, para restringir o grande fluxo de ar;
3. Realizou-se três medições uma, após decorrido trinta minutos e as outras duas cada dez minutos.

Apesar de ser um equipamento que sua temperatura de trabalho seria de 32 à 42 graus Celsius, é fundamental que tenha uma acurácia em várias faixas de temperatura.

Os ambientes utilizados para coleta de dados são:

- Interior de uma geladeira;
- Uma sala com ar condicionado;
- Uma sala com a temperatura ambiente;
- Uma sala com grande incidência de sol.

Após a aquisição de dados os valores obtidos foram:

	Tipo de termômetro	Interior de um geladeira	Sala com ar condicionado	Sala com temperatura ambiente	Sala com incidência de sol
Primeira medição	Mercúrio	11,50	20,50	25,00	37,25
	Protótipo	10,87	20,43	25,31	38,94
Segunda medição	Mercúrio	11,00	20,75	25,00	38,50
	Protótipo	10,69	20,93	25,19	39,63
Terceira medição	Mercúrio	11,00	20,50	25,50	39,00
	Protótipo	11,19	21,13	25,69	39,32

4.4 Programação do Microcontrolador

Para gerenciar as solicitações do aplicativo e a obtenção do valor de temperatura foi desenvolvido um software para o ESP utilizando o Arduino IDE.

O trecho abaixo, demonstra a função responsável por gerenciar as requisições originárias do aplicativo:

```
1 // [Gerencia as requisicoes] =====
2 void verireq() {
3   WiFiClient client = server.available(); // Verifica se ha solicitacao
4   if (!client) { // Se nao houver ele retorna para a funcao loop
5     return;
6   }
7   Serial.println("Novo acesso!");
8   client.setTimeout(1000); // Tempo para realizar a conexao
9
10  while (client.available()) { // Enquanto houver comunicacao com o cliente
11    client.read(); // funcao responsavel por enviar os dados de temperatura
12  }
13  client.print("HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n\r\n"); //
    Configura texto em HTML
14  client.print(temp); // Variavel de temperatura
15 }
```

Nesta programação, que é apresentada na íntegra no apêndice A, é possível observar que a requisição ao TMP102 e cálculo do valor final de temperatura é realizado em quatro funções distintas, uma sendo executada a cada loop, pois ao fazê-lo em apenas uma vez, corre-se o risco do mesmo não responder a solicitação enviada pela aplicação do smartphone, por se tratar de um processo extenso.

5 CONCLUSÃO

Ao realizar o projeto, observou-se que o sistema atuou de forma prática e eficiente, todos os componentes funcionaram corretamente e tiveram um desempenho satisfatório, atingindo assim o objetivo inicial.

A partir deste projeto foi possível verificar a falta de informação e a necessidade de orientações aos pais e cuidadores acerca dos sintomas da febre e de seus causadores.

Por se tratar de um primeiro protótipo, há algumas possíveis melhorias para uma segunda versão:

1. O protótipo foi feito utilizando o Esp 01, porém poderia ser utilizado outros microcontroladores da família ESP que contenham a comunicação I2C e mais pinos, possibilitando utilizar toda a capacidade do sensor;
2. O consumo energético não foi levado em conta na montagem do processo do termômetro, para isso seria necessário mensurar a potencia necessária para dimensionar um circuito de alimentação, além de aplicar a função sleep no ESP para aumentar sua autonomia.
3. A ergonomia não teve muita relevância nessa versão, porém para um experimento empírico será necessário melhor-lo neste aspecto;
4. O protótipo atual apenas se conecta em uma rede padrão, definida na programação, o que atrapalha realizar alguns teste fora do alcance desta rede.

O projeto como um todo, foi muito satisfatório, mesmo se tratando de um protótipo inicial, verificou-se diversos fatores que necessitam de aprimoramento para uma possível versão comercial.

Referências

- Braga, N. C. (2018). Como funcionam os sensores de temperatura (art764).
- Bricks, L. F. (2006). Tratamento da febre em crianças. *Pediatrics. São Paulo*, 28(3):155–8.
- Casanova, C., Reis, S., and Prata, A. (2014). Atuação dos pais na situação febril dos filhos. *I Jornadas do Mestrado de Enfermagem de Saúde Infantil e Pediatria: Livro de resumos*, pages 12–13.
- de Araujo, R. B. (2003). Computação ubíqua: Princípios, tecnologias e desafios.
- de Oliveira, S. (2017). Internet das coisas com esp8266, arduino e raspberry pi.
- Ferreira, H. G. C. (2014). Arquitetura de middleware para internet das coisas.
- Machado, B. M., Cardoso, D. M., Paulis, M. d., Escobar, A. M. d. U., Gilio, A. E., et al. (2009). Febre sem sinais localizatórios: avaliação de um protocolo de atendimento. *Jornal de Pediatria*, 85(5):426–432.
- Magni, Ana Maria e Scheffer, D. K. e. B. P. Efeito antipirético do ibuprofeno e da dipirona em crianças febris. 87.
- Martins, N. A. (2005). Sistemas microcontrolados. *Uma abordagem com o Microcontrolador PIC 16F84*. Editora Novatec Ltda.
- Murahovschi, J. (2003). A criança com febre no consultório. *Jornal de Pediatria*.
- Oliveira, R. R. (2017). *USO DO MICROCONTROLADOR ESP8266 PARA AUTOMAÇÃO RESIDENCIAL*. PhD thesis, Universidade Federal do Rio de Janeiro.
- Pires, D. P. L., Afonso, J. C., and Chaves, F. A. B. (2006). Do termoscópio ao termômetro digital: quatro séculos de termometria. *Química Nova*, 29(6):1393.
- Salgado, M. (2014). Critérios de alerta na criança com febre: texto para pais. *Acedido em http://areadospais.asic.pt/article_files/febre_sinais_alerta.pdf*.
- SILVA, R. M., CAPANEMA, F. D., GONÇALVES, L. A. O., ROCHA, R. L., et al. (2018). Febre infantil e seu manejo pelos pais: Análise quantitativa. *Revista Brasileira de Ciências da Saúde*, 22(2):117–124.
- Simões, A. L. B. and De Martino, M. M. F. (2007). Variabilidade circadiana da temperatura oral, timpânica e axilar em adultos hospitalizados. *Revista da Escola de Enfermagem da USP*, 41(3):485–491.

Apêndice A

Abaixo é apresentado a programação completa e comentado do software desenvolvido na plataforma arduino para o ESP 01.

```
1  #include <ESP8266WiFi.h> //Biblioteca responsavel pela conexao do Esp
    com a rede
2  #include <Wire.h> //Biblioteca responsavel pela comunicacao I2C
3
4  //Funcoes=====
5  void iniWiFi();
6  void verWiFi();
7  void verireq();
8  void tmrsensor();
9  void calctmp01();
10 void calctmp02();
11 void calctmp03();
12 void trterro();
13 void reqi2c();
14 void zrvar();
15
16 //Variaveis=====
17 const char ssid = "Esp"; //Nome da rede
18 const char password = "senha123"; //Senha da rede
19 const int16_t I2C_SLAVE = 0x48; //Endereco do TMP102 para a comunicacao
    I2C
20 boolean auxver = false; //Variavel auxiliar da funcao verWiFi
21 int msb; //Byte mais significativo da comunicacao I2C e auxiliar das
    funcoes para o calculo da temperatura
22 int lsb; //Byte menos significativo da comunicacao I2C e auxiliar das
    funcoes para o calculo da temperatura
23 int bimsb[12]; //Vetor com bits mais signifiativos
24 int bilsb[4]; //Vetor com bits menos signifiativos
25 int pot = 0; // Variavel utilizada no calculo da potencia
26 float temp = 0; //Variavel da temperatura
27 int i; //Variavel auxiliar das funcoes para o calculo da temperatura
28 int k = 4; //Variavel gerenciador das funcoes
29
30 WiFiServer server(80); //Inicializacao do Esp em modo servidor
31
32 //=====
33 void setup() {
34   Serial.begin(115200); //iniciando a comunicacao serial
35   Wire.begin(0,2); //iniciando a comunicacao I2C utilizando o GPIO 0 como
    SDA e o 2 como SCL
```

```
36 delay(10);
37
38 //Calcula o valor de temperatura ao iniciar
39 reqi2c();
40 calctmp01();
41 calctmp02();
42 calctmp03();
43
44 iniWiFi(); //funcao para iniciar a conexao wifi
45 }
46
47 //=====
48 void loop() {
49   if (WiFi.status() != WL_CONNECTED || auxver == false) { //Verifica se o
       Esp esta conectado ou o valor da variavel auxver
50   verWiFi(); //Chama a funcao
51   }
52   verireq();//Chama a funcao
53
54   switch (k) {//Verifica o valor da variavel k e dependendo deste valor
       ele chama a funcao correspondente
55   case 0: calctmp01(); break;
56   case 1: calctmp02(); break;
57   case 2: calctmp03(); break;
58   default: reqi2c(); break;
59   }
60   //envia o valor da temperatura para o monitor serial.
61   Serial.print("Temperatura:");
62   Serial.println(temp, 3);
63   delay(1000);
64   }
65
66   //[Inicia a conexao com a rede]=====
67   void iniWiFi() {
68     Serial.println();
69     Serial.println();
70     Serial.print("Conectando a rede ");
71     Serial.println(ssid);
72     WiFi.begin(ssid, password);//Conecta na rede
73
74     while (WiFi.status() != WL_CONNECTED) { //Verifica se a conexao foi
       realizada
75     delay(500);
76     Serial.print(".");
77     }
78     server.begin(); //inicializa do servidor
79   }
```

```

80
81 // [Verifica a conexao com a rede]=====
82 void verWiFi() {
83   if (WiFi.status() == WL_CONNECTED) { // Verifica se ha conexao
84     Serial.println("");
85     Serial.println("O seu ESP esta conectado!");
86     Serial.print("Endereco IP: ");
87     Serial.println(WiFi.localIP());
88     server.begin(); // inicializa do servidor
89     Serial.println("Servidor iniciado!");
90     auxver = true; // Altera o valor da variavel para que a mensagem nao
        repita contantemente
91   }
92   else {
93     Serial.println("");
94     Serial.println("O seu ESP nao se consegue se conectar...");
95     auxver = false; // Altera o valor da variavel para repetir a mensagem da
        conexao realizada
96   }
97 }
98
99 // [Gerencia as requisicoes]=====
100 void verireq() {
101   WiFiClient client = server.available(); // Verifica se ha solicitacao
102   if (!client) { // Se nao houver ele retorna para a funcao loop
103     return;
104   }
105   Serial.println("Novo acesso!");
106   client.setTimeout(1000); // Tempo para realizar a conexao
107
108   while (client.available()) { // Enquanto houver comunicacao com o cliente
109     client.read(); // funcao responsavel por enviar os dados de temperatura
110   }
111   client.print("HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n\r\n"); //
        Configura texto em HTML
112   client.print(temp); // Variavel de temperatura
113
114 }
115
116 // [Calculo da temperatura - parte 1 ]=====
117 void calctmp01() {
118
119   // Converte o byte mais significativos de decimal para binario e os
        aloca em oito celulas do vetor bimsb
120   for (i = 4; i < 12; i++) {
121     bimsb[i] = msb % 2;
122     msb = msb / 2;

```

```
123
124 }
125 // Converte parte do byte menos significativos de decimal para binario e
    os aloca nas quatro celulas do vetor bilsb
126 for (i = 0; i < 4; i++) {
127     bilsb[i] = lsb % 2;
128     lsb = lsb / 2;
129 }
130 k++; //Altera o valor da variavel k para passar para o proximo passo
131 }
132
133 //[Calculo da temperatura - parte 2 ]=====
134 void calctmp02() {
135     // Converte o resto do byte menos significativos de decimal para binario
        e os aloca nas quatro celulas restantes do vetor bimPsb
136     for (i = 0; i < 4 ; i++) {
137         bimsb[i] = lsb % 2;
138         lsb = lsb / 2;
139     }
140     // Converte os doze bits mais significativos de binario para decimal
141     msb = 0;
142     pot = 1;
143     for (i = 0; i < 12; i++) {
144         msb = msb + (pot * bimsb[i]);
145         pot = pot * 2;
146     }
147 }
148 // Converte os quatro bits restantes de binario para decimal
149 lsb = 0;
150 pot = 1;
151 for (i = 0; i < 4; i++) {
152     lsb = lsb + (pot * bilsb[i]);
153     pot = pot * 2;
154 }
155 }
156 k++; //Altera o valor da variavel k para passar para o proximo passo
157 }
158
159 //[Calculo da temperatura - parte 3 ]=====
160 void calctmp03() {
161     //Verifica valor de lsb, para realizar a equacao correta.
162     if (lsb == 0) {
163         temp = (msb * 0.0625);
164     }
165     else {
166         temp = (msb * (0.0625 / lsb));
167     }
```

```
168 k++; //Altera o valor da variavel k para passar para o proximo passo
169 zrvar(); //funcao para zerar as variaveis
170 }
171
172 // [Requisita e recebe os dados do sensor ]=====
173 void reqi2c()
174 { Wire.requestFrom(I2C_SLAVE, 2); //requisita 2 Bytes para o TMP102
175   while (Wire.available()) { //Enquanto houver comunicacao
176     msb = Wire.read(); // Recebe o valor do byte mais significativo
177     lsb = Wire.read(); // Recebe o valor do byte menos significativo
178
179     k = 0; //Altera o valor da variavel k para passar para o proximo passo
180
181
182   }
183 }
184
185 // [Zera os valores da variaveis ]=====
186 void zrvar() {
187   msb = 0;
188   lsb = 0;
189   bimsb[12] = 0;
190   bilsb[4] = 0;
191   pot = 0;
192   i = 0;
193 }
```