

UNIVERSIDADE FEDERAL DO ABC – UFABC

CURSO DE GRADUAÇÃO EM ENGENHARIA DA INFORMAÇÃO

FERNANDO HENRIQUE GOMES FAGGI

**PROTOTIPO DE CLIMATIZAÇÃO EM UMA
CASA INTELIGENTE**

SANTO ANDRÉ/SP

2016

FERNANDO HENRIQUE GOMES FAGGI

PROTOTIPO DE CLIMATIZAÇÃO EM UMA CASA INTELIGENTE

Trabalho desenvolvido durante a disciplina de *Trabalho de Graduação*, e apresentado no curso de graduação em Engenharia da Informação da Universidade Federal do ABC - UFABC.

Orientador: Prof. Dr. Marcelo Bender Perotoni

SANTO ANDRÉ/SP

2016

Conteúdo

1. Introdução	5
Casas Inteligentes.....	5
Conforto térmico.....	7
Consumo de energia	8
Internet das Coisas	8
2. O Sistema de Controle.....	10
2.1 Objetos de controle.....	10
2.2 Dispositivos de atuação.....	11
2.3 Dispositivos de detecção	11
2.4 Dispositivos de Comando	12
2.5 Dispositivos de Controle	12
2.6 Tipos de controle.....	13
2.6.1 Controladores de duas posições ou <i>on-off</i>	13
2.6.2 Controle Proporcional	13
2.6.3 Ação de Controle integral:	13
2.6.4 Ação controle proporcional-integral (PI):	14
2.6.5 Ação controle proporcional-derivativo (PD):	14
2.6.6 Ação controle proporcional-integral-derivativo (PID):.....	14
2.7 Técnicas de controle inteligente	15
2.8 Arquitetura do sistema de controle	15
2.9 Controle de zona	17
2.10 Classificação de Sistemas	18
3. Protótipo.....	20
3.2 O Arduino	20
3.3 <i>Shields</i>	22
3.3.1 <i>Ethernet Shield</i>	22
3.4 Sistema de Monitoramento (GUI).....	23
3.5 Sensores	24
3.5.1 <i>Sensor de Temperatura e Umidade DTH11</i>	24
3.5.2 <i>Sensor PIR</i>	26
4. Metodologia	29
4.2 Materiais utilizados	30

4.3	Conversões do Sinal.....	30
4.4	Arquitetura do Sistema.....	32
4.5	Diagrama de blocos.....	36
5.	Resultados.....	39
5.2	Funcionamento do Sistema.....	39
5.3	Sensores e atuadores.....	42
5.4	Cenário.....	42
5.5	Análises de Resultados.....	46
6.	Conclusões.....	50
7.	Referências Bibliográficas.....	52
8.	Anexos.....	54
8.2	Esquemas Elétricos.....	54
8.2.1	Esquema Elétrico do <i>Ethernet Shield</i>	54
8.2.2	Esquema de ligação dos sensores.....	55
8.2.3	Códigos de programação.....	56

1. Introdução

Casas Inteligentes

Não há como falar de tecnologia sem realizar uma breve análise sobre o surgimento da mesma, ainda mais no que tange às atividades de automações, foco principal deste trabalho. Os primeiros equipamentos que apresentam verdadeiras automações estavam e permanecem até hoje no uso mais corriqueiro do cotidiano da sociedade, como é o caso de geladeiras, fogões, máquinas de lavar, e/ou utensílios domésticos de forma geral. Estes trouxeram uma significativa melhora na rotina doméstica, tornaram processos antes longos, demorados e dispendiosos em processos práticos e ágeis, conseqüentemente proporcionando menos stress, cansaço e insatisfação aos usuários, dando-os também a oportunidade de dedicarem maior tempo a atividades de lazer. Os benefícios obtidos pelo uso da automação de residências foram tão notoriamente significativos que se tornou um campo relevante para atuação, uma vez que representa uma fatia extensa do mercado.

Pouco a pouco outros dispositivos eletrônicos foram também surgindo, como Smart TV's, DVD's, Vídeo Games, rádios e computadores, que a princípio vinham para complementar atividades de lazer domésticas, no entanto, diante de constantes melhoramentos e aplicações, foram tornando-se além de populares, fontes de entretenimento, até atingirem o mercado de tal forma a se estabelecerem como itens indispensáveis.

As pessoas assistem TV ou ouvem música para relaxar, para gastar o seu tempo de lazer e repouso apreciando um filme ou sua banda favorita. O único problema que existia inicialmente era a necessidade de locomover-se até o aparelho, a fim de alterar algumas configurações diretamente no dispositivo como volume, canal, etc. Logo o controle remoto foi introduzido mais uma vez com o papel facilitador e satisfatório ao consumidor, e hoje as pessoas não podem imaginar operar suas TVs na falta deste, tendo que abrir mão do conforto da poltrona para mudar de canal, ainda que por um curto espaço de tempo e distância. É natural então chegar à conclusão que a força motriz para o progresso da tecnologia *at home* que promove hoje, no Brasil, um giro imenso de capital fornecido do mercado consumidor aos desenvolvedores e fabricantes, é a comodidade, praticidade e conforto que proporcionam aos usuários.

Dia a dia são apresentadas novas tendências tecnológicas ao mercado, e uma delas, que é o objetivo deste estudo, são os dispositivos que podem ser instalados em qualquer residência promovendo o controle remoto de diversos dispositivos locais, como, por exemplo, sistema de iluminação e climatização, concretizando assim a ideia atrelada ao conceito de tecnologia inteligente.

Uma casa inteligente utiliza sistemas de automação para gerenciar e integrar subsistemas, monitorando e controlando funções locais, como temperatura, iluminação, sistemas de segurança, controle de janelas e portas, entre outros, proporcionando conforto e praticidade, de forma simples e ao mesmo tempo sofisticada. O conforto térmico destaca-se como elemento de essencial no atendimento aos requisitos de uma casa inteligente e evidencia a necessidade de uma efetiva técnica de modelagem e simulação [1].

Conceitualmente, uma *casa inteligente* é aquela que incorpora sistemas avançados de automação de modo a prover aos habitantes um sofisticado monitoramento e controle sobre diversas funções do local. Por exemplo, em uma casa inteligente pode haver o autocontrole de itens como iluminação, temperatura, multimídia, sistemas de segurança, janelas e portas, além de diversas outras funções.

O termo original vem do inglês *smart homes*. Propondo o sistema que faz uso de diversas tecnologias que, quando interagem entre si, fornecem aos usuários e/ou proprietários uma espécie de resumo inteligente com informações da residência pelo uso do monitoramento de varias funções conforme ilustrado na Figura 1.

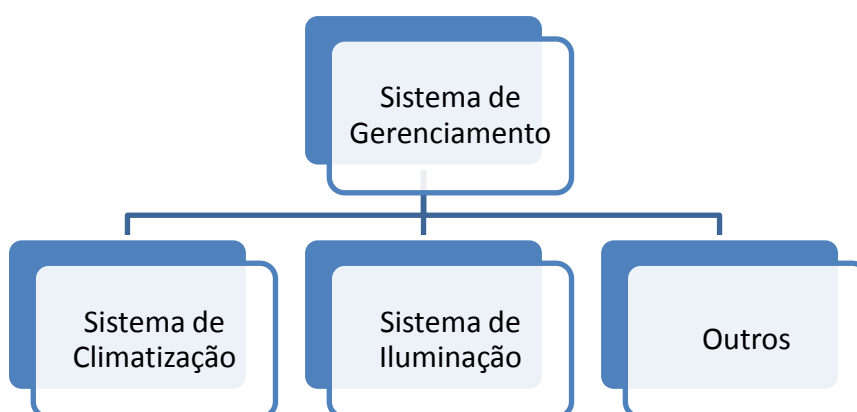


Figura 1.1– Integração de sistemas em uma casa inteligente.

Pelo uso desse monitoramento a casa inteligente pode ativar e desativar sistemas, podendo até mesmo executar funções simples e cotidianas como regar as plantas ou alimentar animais de estimação na ausência dos moradores.

Conforto térmico

O ser humano é naturalmente homeotérmico. Seu organismo possui uma temperatura interna aproximadamente constante, da ordem de 37°C. A manutenção da temperatura interna relativamente constante, em ambientes cujas condições termo higrométricas são as mais variadas e variáveis, se faz por intermédio de seu aparelho termorregulador que comanda a redução ou aumento das perdas de calor pelo organismo por meio de alguns mecanismos de controle. O organismo humano gera energia por meio do metabolismo, de qual parte é transformada em trabalho e o restante é transmitido ao ambiente por meio dos seguintes mecanismos: convecção, radiação, evaporação por difusão e através da respiração [2].

De acordo com a norma [3], “conforto térmico” para uma pessoa pode ser definido como a satisfação com o ambiente no qual ela se encontra o que significa que a pessoa em questão não prefere um ambiente mais frio ou mais quente. Para os climas mais quentes, 25°C como temperatura ótima, podendo variar entre 23°C e 27°C sendo que esses valores são aplicados para velocidade do ar de 0,5 m/s e umidade relativa entre 40 e 60%, vestimenta de meia estação, pessoa sentada, ocupação sedentária e temperatura radiante média igual à temperatura do ar.

Aquecimentos ou resfriamentos inadequados ou não uniformes podem contribuir para o aumento da sensibilidade dos ocupantes a contaminantes, pois altas temperaturas e umidade aceleram a liberação de componentes orgânicos voláteis, como formaldeídos, de materiais da casa e mobílias.

Com a diminuição da umidade, aumenta o ressecamento interno do nariz diminuindo a quantidade de muco, o que favorece a entrada de micro-organismos. Fungos, ácaros e alguns tipos de vírus, como o da poliomielite, preferem ambientes muito úmidos. Alto teor de umidade, acima de 70%, favorece a proliferação de bactérias. O teor de umidade mais favorável ao home é de 50%, onde diminui a incidência de doenças respiratórias.

Consumo de energia

De acordo com a Light, empresa concessionária administradora do fornecimento de energia elétrica em 31 municípios do estado do Rio de Janeiro, o uso de ar condicionado pode representar um aumento até 166% do gasto com energia elétrica nos domicílios. Pelo cálculo da empresa, a utilização de ar condicionado durante 15 dias do mês representa um aumento de 120 kWh, se utilizado o mês todo o consumo passa de 240 kWh, considerando que outros eletrodomésticos de uma casa gastam em torno de 144 kWh por mês, o gasto com ar condicionado representa 45% do gasto total de uma residência em 15 dias. Ou seja, uma residência que utilizar o ar condicionado todos os dias do mês gastará, em média 384 kWh, 166% a mais do que gastaria se não tivesse o ar condicionado, esses dados podem variar de acordo com a potência e modelo do ar condicionado. A necessidade contínua de desenvolver novas formas de melhoria de recursos e de economia de energia propiciou a evolução dos sistemas de ar condicionado, integrando-os ao conceito de edifícios e casas inteligentes.

Internet das Coisas

A ideia de conectar objetos é discutida desde 1991, quando a conexão TCP/IP e a internet de hoje começou a se popularizar, o termo Internet das Coisas (Internet of things) foi proposto por Kevin Ashton em 1999.

Atualmente a Internet das coisas gira em torno de um aumento de máquina para máquina de comunicação, construída sobre computação em nuvem e redes de sensores para coletas de dados que é móvel, virtual e de conexão instantânea. Porém todas as informações coletadas por todos os sensores no mundo não vale muito se não houver uma infraestrutura no local para analisá-lo em tempo real. Aplicativos baseados em nuvem são a chave para o uso de dados avançadas. A Internet das coisas não funciona sem aplicativos baseados em nuvem para interpretar e transmitir os dados provenientes de todos esses sensores. A nuvem é o que permite que os aplicativos para ir trabalhar para você a qualquer hora, em qualquer lugar. Portanto, agora temos de monitoramento sensores e rastreamento de todos os tipos de dados; temos aplicativos baseados em

nuvem traduzindo esses dados em informações úteis e transmiti-lo às máquinas no terreno, permitindo, respostas em tempo real para dispositivos móveis. E, assim, tornar-se pontes para desenvolvimentos de infinitos projetos, como uma casa inteligente.

2. O Sistema de Controle

Como principais parâmetros a serem controlados por um sistema de ar condicionado podem-se destacar a temperatura, velocidade relativa, umidade e concentração de CO₂, existência de odores dentre outras. Cabe ao sistema de ar condicionado manter a combinação adequada das variáveis, de forma atender as necessidades de conforto térmico do usuário ou para funcionamento ideal de equipamento [1].

Considerando o sistema de ar condicionado como um sistema produtivo, o qual o produto final é o ar condicionado, pode-se adotar o modelo apresentado na figura 2.1, para a classificação de diversas partes do sistema de ar condicionado [13].

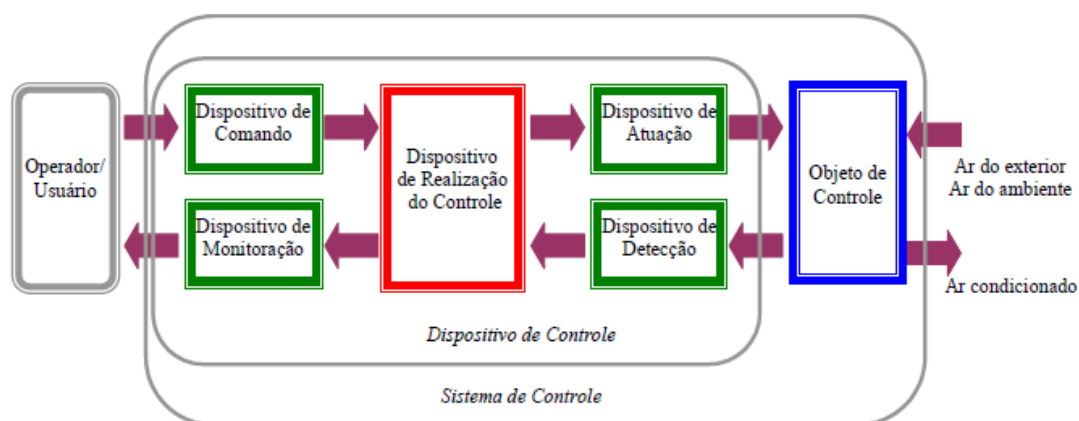


Figura.2.1 – Diagrama do sistema de controle

2.1 Objetos de controle

Para climatizar uma casa inteligente é viável o uso de ar condicionado de expansão direta, que faz a troca de calor diretamente entre o fluido refrigerante e o ar a ser direcionado. A serpentina de troca térmica trabalha como evaporador do ciclo de refrigeração do fluido refrigerante. Dentre os equipamentos que utilizam esta tecnologia podem-se destacar os aparelhos convencionais de ar condicionado de janela, os

aparelhos de ar condicionado tipo *split* e *self-contained*, que podem ter insuflado o ar direto no ambiente ou trabalhar com redes de distribuição. O objeto de controle no caso do ar condicionado residencial será o fluxo de ar distribuído pelo ambiente dos quais se deseja controlar propriedades como temperatura e vazão.

2.2 Dispositivos de atuação

São responsáveis pela modificação do estado do objeto de controle a partir do sinal recebido pelo dispositivo de realização do controle, como principais para uma casa inteligente, sendo alguns exemplos:

- Válvulas de bloqueio: utilizadas para bloquear o fluxo de ar.
- Ventiladores: realizam o transporte do ar através dos dutos.
- Dampers: Registros que regulam a passagem de ar e são utilizados para regular a quantidade de ar insuflado em sistemas VAV (*Variable Air Volume*). Um modelo de damper pode ser observado na figura 2.2.1.



Figura 2.2.1 – Damper controlável por sistema wireless.

2.3 Dispositivos de detecção

Os dispositivos que fazem a detecção do estado do sistema são os sensores e/ou transdutores. Sendo o sensor que fará a medida das variáveis de controle, sendo os principais sensores utilizados em sistema de ar condicionado:

- Sensor de temperatura: utilizado para medir a temperatura do ar nas zonas de controle.
- Sensores de umidade: utilizados para medir a umidade caso se utilize como variável no sistema.
- Sensores de Pressão: podem ser utilizados para se verificar a velocidade dos ventiladores em uma zona de controle.
- Sensores de presença: podem ser utilizados para verificar a existência no local de habitantes.

Vários sistemas de localização têm sido desenvolvidos para reconhecer a localização de um habitante em uma casa inteligente. Em geral, em um sistema indoor a localização tem sido classificada em três tipos, estabelecidos de acordo com o tipo de tecnologia da medida: triangulação, análise de cena e pelo método de aproximação. [5]

De acordo com [5], ao se criar uma matriz que usa sensores do tipo PIR (*Passive Infrared Sensor*), pode-se distribuí-los no teto de forma a organizá-los para que as áreas de detecção dos sensores adjacentes se sobreponham. Ao combinar as saídas dos vários sensores PIR, o sistema é capaz de localizar um habitante com razoável grau de precisão.

2.4 Dispositivos de Comando

Os dispositivos de comando variam de acordo com o tipo de sistema que podem ser representados por botões, chaves e outras interfaces dos usuários com o sistema de ar condicionado como, por exemplo, comandos por softwares de computador e até via *smartphones*.

2.5 Dispositivos de Controle

Os dispositivos de controle fazem a recepção dos sinais recebidos dos dispositivos de detecção e dos dispositivos de comando, comparando os valores recebidos com o valor desejado; processando os dados e enviado esses sinais para os

atuadores do sistema, podendo também enviar esses sinais para um sistema de monitoração.

2.6 Tipos de controle

Um controlador automático compara o valor real de saída com a entrada de referência (valor desejado), determina o desvio e produz um sinal de controle que reduzirá o desvio a zero ou a um valor pequeno. A maneira pela a qual o controlador produz o sinal de controle é chamada ação de controle [6].

Em geral para um sistema de ar condicionados os controladores podem ser classificados, de acordo com suas ações de controle, em:

2.6.1 Controladores de duas posições ou *on-off*.

O elemento atuante tem apenas duas posições fixas, que são, em muitos casos, simplesmente ligado e desligado.

2.6.2 Controle Proporcional

Para um controlador com ação de controle proporcional, a relação entre a saída do controlador $u(t)$ e o sinal de erro atuante $e(t)$ é:

$$u(t) = K_p e(t)$$

Onde K_p é o ganho proporcional

2.6.3 Ação de Controle integral:

Um controlador com ação de controle integral, o valor da saída $u(t)$ do controlador é modificado a uma taxa de variação proporcional ao sinal de erro atuante $e(t)$, ou seja:

$$u(t) = K_i \int_0^t e(t) dt$$

Onde K_i é uma constante ajustável. A função de transferência de um controlador integral é:

$$\frac{U(s)}{E(s)} = \frac{K_i}{s}$$

2.6.4 Ação controle proporcional-integral (PI):

A ação de controle proporcional integral é definida por:

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt$$

Onde K_i é o ganho integral e T_i é chamado tempo integrativo.

2.6.5 Ação controle proporcional-derivativo (PD):

A ação de controle proporcional derivativo é definida por:

$$u(t) = K_p e(t) + K_p T_d \frac{de(t)}{dt}$$

Onde T_d é chamado tempo derivativo e $\frac{de(t)}{dt}$ é a derivada do erro em relação ao tempo.

2.6.6 Ação controle proporcional-integral-derivativo (PID):

A combinação das ações de controle proporcional, de controle integral e de controle derivativo é denominada ação de controle proporcional-integral-derivativo.

Essa ação combinada tem as vantagens individuais de todas as umas das três ações de controle. A equação de um controlador com essas ações combinadas é dá por:

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p T_d \frac{de(t)}{dt}$$

Onde T_d é chamado tempo derivativo K_i é o ganho integral, T_i é chamado tempo integrativo e K_p é o ganho proporcional.

No entanto a adição do termo derivativo também faz com que o controlador seja sensível a ruídos, assim, a maioria das malhas de controle dos sistemas de ar condicionado obtém na pratica resultados mais satisfatórios com um controlador do tipo Proporcional-Integral [1].

2.7 Técnicas de controle inteligente

Além das técnicas de controle descritas acima podem ser utilizadas técnicas derivadas de inteligência artificial, lógica fuzzy e até uso de redes neurais.

2.8 Arquitetura do sistema de controle

O aumento do custo da energia elétrica tem motivado o desenvolvimento de novas tecnologias para a redução do consumo. Para sistemas de ar condicionado, tem sido introduzida tecnologias como sistemas de controle de volume de ar variável (VAV) sistema de controle com malha fechada.

Os sistemas de controle com realimentação são, com frequência, denominados também sistemas de controle de malha fechada. Na pratica, os termos controle com realimentação e controle de malha fechada são usados indistintamente. Em um sistema de controle de malha fechada, o sinal de erro atuante, que é diferença entre o sinal de entrada e o sinal de realimentação (que pode ser o próprio sinal de saída ou uma função do sinal de saída e suas derivadas/e ou integrais), realimenta o controlador, de modo a minimizar o erro e acertar a saída do sistema ao valor desejado. O termo

controle de malha fechada sempre implica a utilização do controle com realimentação para reduzir erros do sistema. [1]

Existem diversas variações sobre a forma de controle da temperatura dos ambientes climatizados. O sistema de controle por zonas, que mantém toda a área atendida (zonas) por um mesmo ar condicionado sob a mesma temperatura tem em geral, um sensor de temperatura instalado no duto de retorno, próximo ao ar condicionado, após a parte que mistura o ar de todos os ambientes. A partir do sinal emitido por este sensor, o fluxo de ar frio é ajustado e a temperatura do ar insuflado é ajustada. Este sistema é conhecido como sistema de controle de temperatura VAC (volume de ar constante), uma vez que não há variação do fluxo de ar insuflado, somente na temperatura do ar insuflado.

Em sistemas de controle individualizados por ambiente, na qual a temperatura pode ser regulada de forma independente, cada ambiente tem seu próprio sensor de temperatura, cujo sinal determina o posicionamento do damper desta forma se introduz um controle do volume de ar (VAV), instalado no duto de ar condicionado.

Neste caso, o ajuste da temperatura é feito por meio da regulação do volume de ar insuflado sobre o ambiente, como pode ser observado no exemplo da figura 2.8.1:

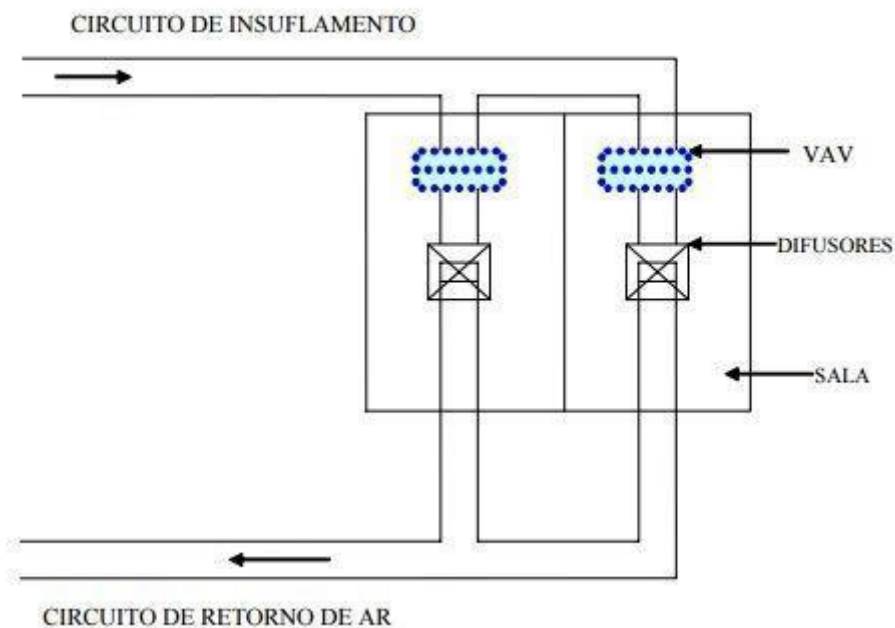


Figura 2.8.1 - Esquema de operação de um controle de volume de ar (VAV)

O sistema de controle de temperatura ambiente utilizando o método de volume de ar variável e a variação da velocidade nas saídas de ar frio são exemplos de melhoria introduzidos em virtude das novas formas de monitoração e controle.

O sistema de ar condicionado pode ser integrado aos demais sistemas, como por exemplo, pode-se citar a integração a sistemas de controle de acesso, o que proporciona o acionamento do sistema de ar condicionado somente quando é verificada a presença de habitantes nos ambientes, ou como um sistema de combate a incêndio de forma controlar o fluxo de ar para as zonas onde esteja ocorrendo o incêndio, reduzindo assim o volume de ar que venha a alimentar o fogo, ou em zonas adjacentes à de ocorrência do incêndio, de forma a impedir a propagação da fumaça.

2.9 Controle de zona

Por meio do controle de zona é possível controlar as propriedades de temperatura e umidade em função do tipo de sistema de ar condicionado.

Para sistemas do tipo volume de ar variável, o controle das posições dos “*dampers*” de insuflamento de ar é realizado por meio da temperatura de do ambiente e geralmente o controle é realizado pelo controle proporcional.

O sistema é caracterizado pela alta velocidade de descarga misturando-se ao ar da sala, no chamado movimento de ar primário, que ocorre fora da zona ocupada. O movimento do ar secundário ocorre na zona ocupada, com velocidade do ar mais baixa, de 0,25 m/s, segundo as normas de conforto. O tipo de difusor e a velocidade de descarga determinam a taxa de mistura [3].

Se um sistema de condicionamento de ar com insuflamento pelo teto for bem projetado, vai garantir uma mistura do ar quase perfeita para resfriamento e alcançar uma eficiência de troca de ar em quase 100% [3].

A escolha do tipo de difusor e sua localização são importantes para garantir tanto o conforto térmico quanto a renovação eficiente de poluentes internos e são

influenciados pelo layout interno do ambiente e pela presença de fontes de calor em locais específicos, que podem causar estratificação vertical da temperatura [3].

2.10 Classificação de Sistemas

Os sistemas de ar condicionado são sistemas dinâmicos, e como tais podem ser classificados com base na caracterização de suas variáveis de estado em relação ao tempo.

Os sistemas a eventos discretos (SEDs) caracterizam-se pelas transições instantâneas entre estados discretos. As variáveis de estado variam abruptamente em determinados instantes. Estes sistemas são em geral baseados em regras e procedimentos definidos pelo homem. Para esse tipo de sistema, o objetivo do controle é a execução de operações caracterizada pela ocorrência de eventos, conforme um procedimento pré-definido [13].

A principal característica dos sistemas de variáveis continua (SVCs) é que as variáveis de estado evoluem de forma continua no tempo em função de fenômeno de natureza, isto é, baseadas em leis físicas. Para sua modelagem, os sistemas de equações diferenciais são ferramentas comprovadamente eficazes. Nestes sistemas, o objetivo básico do controle normalmente é igualar o valor de uma variável de controle a um valor de referência.

A partir destes conceitos, deriva-se a ideia de sistemas híbridos, no qual é necessária a análise tanto do ponto de vista contínuo quanto das interferências proporcionadas pelos eventos discretos. São encontradas características tanto de SEDs como SVCs. Um exemplo prático deste tipo de sistema são os sistemas de ar condicionado, nos quais existem estados discretos, caracterizados pelo status de equipamentos e variáveis contínuas no tempo, como os de combate a incêndio, controle de acesso dentre outros. A característica contínua também é marcante, por meio de variáveis contínuas como temperatura e umidade, com comportamentos regidos por leis físicas, assim como na interação com ambientes interno e externo, conforme ilustrado na figura 2.10.1.

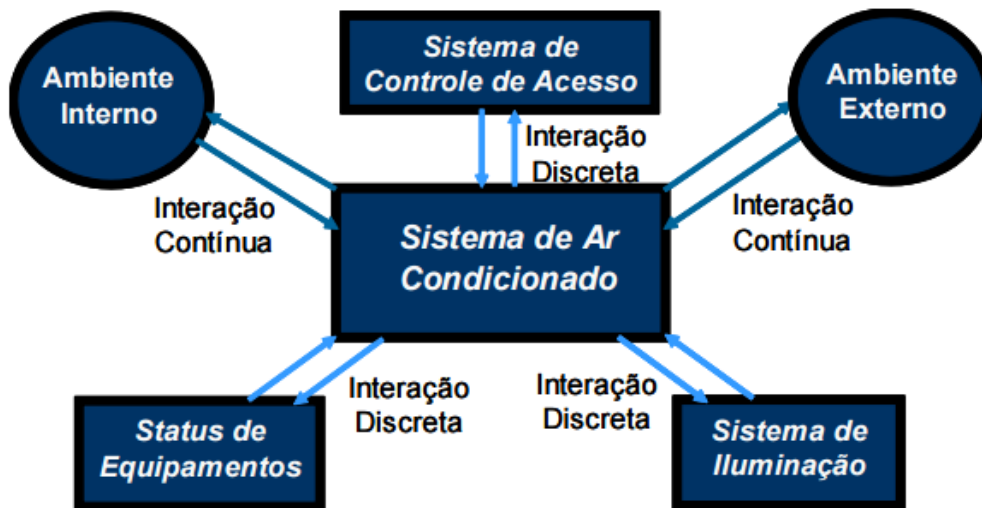


Figura 2.10.1 - Interações do sistema de ar condicionado

3. Protótipo

Como citado na parte 2 Os dispositivos de controle farão a recepção dos sinais recebidos dos dispositivos de detecção e dos dispositivos de comando, comparado os valores recebidos com o valor desejado, processando os dados e enviado esses sinais para os atuadores do sistema o dispositivo que fará esse controle será o Arduino

3.2 O Arduino

Um Arduino é um sistema micro controlado de placa única e um conjunto de software para programá-lo. O hardware consiste em um projeto de hardware livre para o controlador, com um processador Atmel AVR e suporte embutido de entrada/saída. O software consiste de uma linguagem de programação padrão e do boot loader que roda na placa.

Em termos práticos, um Arduino é um pequeno computador que pode ser programador para processar entradas e saídas entre o dispositivo e os componentes externos conectados a ele (figura 3.2.1).



Figura 3.2.1- Arduino Mega [8]

O Arduino é o que chamamos de plataforma de computação física ou embarcada, ou seja, um sistema que pode interagir com seu ambiente por meio de hardware e software.

Por exemplo, um uso simples de um Arduino seria para acender uma luz por certo intervalo de tempo, digamos, 30 segundos, depois que um botão fosse pressionado. Nesse exemplo, o Arduino teria uma lâmpada e um botão conectados a ele.

O Arduino aguardaria pacientemente até que o botão fosse pressionado; uma vez pressionado o botão, ele acenderia a lâmpada e iniciaria a contagem. Depois de contados 30 segundos, apagaria a lâmpada e aguardaria um novo apertar do botão. Essa configuração poderia ser utilizada para controlar uma lâmpada em um closet, por exemplo.

Esse conceito poderia ser estendido pela conexão de um sensor, como um sensor de movimento PIR, para acender a lâmpada quando ele fosse disparado.

O Arduino pode ser utilizado para desenvolver objetos interativos independentes, ou pode ser conectado a um computador, a uma rede, ou até mesmo à Internet para recuperar e enviar dados do Arduino e atuar sobre eles.

Em outras palavras, ele pode enviar um conjunto de dados recebidos de alguns sensores para um site, dados estes que poderão, assim, ser exibidos na forma de um gráfico. [7]

O Arduino utilizado nesse controle Arduino Mega é uma placa de micro controlador baseado no ATmega1280. Ele tem 54 pinos digitais de entrada / saída (dos quais 14 podem ser usados como saídas PWM), 16 entradas analógicas, 4 UARTs (portas seriais de hardware), rodando com um oscilador a cristal estabilizado em 16 MHz além de uma conexão USB.

A programação do Arduino é feita em IDE (ambiente de desenvolvimento integrado) própria, baseada em plataforma Java. No ambiente é feita a transferência de códigos para o micro controlador usando o cabo USB. A linguagem é baseada em C/C++ com funções próprias para o Arduino. Possui biblioteca de funções com várias implementações já programadas. A linguagem é estruturada em duas funções principais: setup e loop. Todo código deve possuir essas duas funções, sendo que o compilador não carrega o programa se alguma das duas não estiver presente.

3.3 Shields

Shields são placas de expansão que podem ser plugados em cima do PCB Arduino estendendo suas capacidades. As diferentes *Shields* seguem a mesma filosofia que o a do kit original - eles são fáceis de montar e com baixo custo.

3.3.1 Ethernet Shield

Ethernet é uma arquitetura de interconexão para redes locais - Rede de Área Local (LAN) - baseada no envio de pacotes. Ela define cabeamento e sinais elétricos para a camada física, em formato de pacotes e protocolos para a subcamada de controle de acesso ao meio (*Media Access Control* - MAC) do modelo OSI. A Ethernet foi padronizada pelo IEEE como 802.3.

O *Ethernet Shield* (figura 3.3.1) conecta o Arduino à internet ou a uma rede local. Ligando este módulo sobre a placa Arduino pode-se conectá-lo a rede com através de uma conexão RJ45 e desenvolver algumas instruções é possível controlar e receber informações do Arduino através da internet ou rede local.

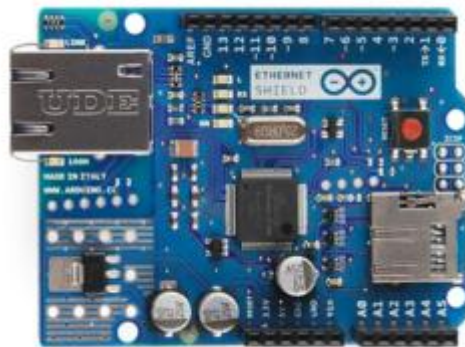


Figura 3.3.1- *Ethernet Shield* [8]

Como vantagem de se utilizar o Arduino, cada elemento da plataforma - hardware, software e documentação - é livremente disponível e de código aberto.

Isto significa que se pode aprender exatamente como ele é feito e usar seu design como ponto de partida para outros circuitos, o esquema elétrico da ethernet shield pode ser analisado na seção de anexos.

Através do Shield de Ethernet e utilizando-se do protocolo de TCP/IP (também chamado de pilha de protocolos TCP/IP) é um conjunto de protocolos de comunicação entre computadores em rede. Seu nome vem de dois protocolos: o TCP (Transmission Control Protocol - Protocolo de Controle de Transmissão) e o IP (Internet Protocol - Protocolo de Internet, ou ainda, protocolo de interconexão). O conjunto de protocolos pode ser visto como um modelo de camadas (Modelo OSI), onde cada camada é responsável por um grupo de tarefas, fornecendo um conjunto de serviços bem definidos para o protocolo da camada superior. As camadas mais altas estão logicamente mais perto do usuário (chamada camada de aplicação) e lidam com dados mais abstratos, confiando em protocolos de camadas mais baixas para tarefas de menor nível de abstração. A placa Arduino pode-se para criar um servidor Web simples. Usando a biblioteca Ethernet, o dispositivo será capaz de responder a uma solicitação HTTP que é um protocolo de comunicação (na camada de aplicação segundo o Modelo OSI) utilizado para sistemas de informação de hipermídia, distribuídos e colaborativos. Ele é a base para a comunicação de dados da *World Wide Web*. Através do shield Ethernet. Ao abrir um navegador e navegando para o endereço IP do Arduino este vai responder com HTML.

3.4 Sistema de Monitoramento (GUI)

Em ciência da computação, uma interface gráfica do usuário ou GUI, é um tipo de interface de que permite que os usuários a interagir com dispositivos eletrônicos por meio de gráficos, ícones e indicadores visuais, como a notação secundário, ao contrário de interfaces baseadas em texto, datilografado etiquetas de comando ou navegação de texto.

Projetando a composição visual e comportamento temporal de uma GUI é uma parte importante da aplicação de software de programação na área de interação humano-computador [9].

Seu objetivo é aumentar a eficiência e facilidade de uso para o projeto lógico de um programa, uma disciplina de design conhecida como usabilidade. Métodos

de design centrado no usuário são usados para assegurar que a linguagem visual introduzida no projeto é bem adaptada às tarefas.

Para se criar um GUI simples e possível de se executar com o Arduino Ethernet Shield pode-se fazer o uso de programação HTML e criar uma GUI que pode ser acessada por qualquer navegador.

HTML é a sigla de *Hypertext Markup Language*, expressão inglesa que significa "Linguagem de Marcação de Hipertexto". Consiste em uma linguagem de marcação utilizada para produção de páginas na web, que permite a criação de documentos que podem ser lidos em praticamente qualquer tipo de computador e transmitidos pela internet.

Para escrever documentos HTML não é necessário mais do que um editor de texto simples e conhecimento dos códigos que compõem a linguagem. Os códigos (conhecidos como *tags*) servem para indicar a função de cada elemento da página Web. Os *tags* funcionam como comandos de formatação de textos, formulários, links (ligações), imagens, tabelas, entre outros.

Os browsers (navegadores) identificam as *tags* e apresentam a página conforme está especificada. Um documento em HTML é um texto simples, que pode ser editado em qualquer editor de Texto e transformado em hipertexto.

3.5 Sensores

3.5.1 *Sensor de Temperatura e Umidade DTH11*

O sensor DTH11 (figura 3.5.1) é distribuído pela empresa D-Robotic, é um sensor de temperatura e umidade que possui um sinal digital na saída. O sensor possui um componente resistivo para medição da umidade, um componente NTC para medir a temperatura e necessita de um micro controlador (MCU) de 8-bits para receber os dados de temperatura e umidade. A figura 3.5.2 mostra a ligação do DTH11 [10].

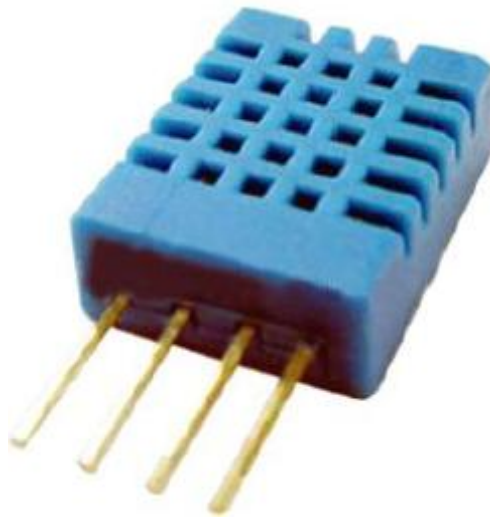


Figura 3.5.1- Sensor DHT11 – Fonte [10]

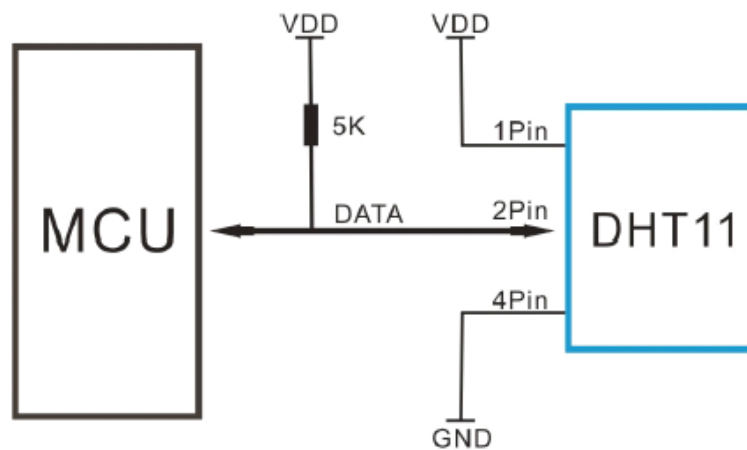


Figura 3.5.2- Aplicação Típica Fonte [10]

Alimentação e Ligação

O sensor DHT11 opera com sua alimentação na faixa de 3-5.5V DC. Quando a alimentação é fornecida ao sensor, este não envia qualquer instrução para o sensor em menos de um segundo, a fim de passar o estado instável. Um capacitor com valor de 100nf pode ser adicionado entre VDD e GND para filtragem da alimentação.

Processo de Comunicação

O processo de comunicação possui um formato de dados chamado Single-Bus, onde é usado para a comunicação e sincronização entre MCU e o sensor de

DHT11. O processo de comunicação dura cerca de 4ms em uma transmissão de dados completa de 40 bits. O formato dos dados pode ser dividido em:

8bits integrantes decimais dados RH + 8bits dados RH + 8bits dados T integrante + 8bits decimal T + a soma de verificação de dados de 8bits = 40 bits [10]

A figura 3.5.3 representa a comunicação entre a MCU e o DTH11:

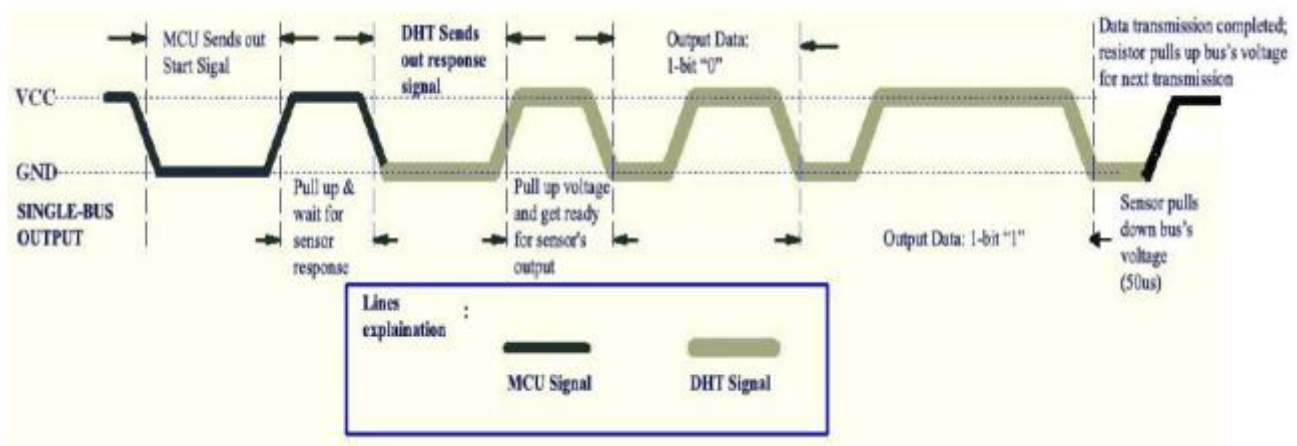


Figura 3.5.3 - Processo de comunicação modo geral [10]

Caso a transmissão de dados esteja correta, a verificação de soma deve ser a última amostra de 8 bits. Quando o MCU envia um sinal de partida, o sensor DHT11 muda do modo de baixo consumo de energia para o modo de funcionamento, à espera de MCU completando o sinal de partida. Uma vez concluída, o sensor envia um sinal de resposta de dados de 40 bits, que incluem a umidade relativa e informações de temperatura para MCU. Os usuários podem optar por recolher (ler) alguns dados. Sem o sinal de partida do MCU, o sensor não dará o sinal de resposta ao MCU. Uma vez que os dados são coletados, o sensor irá mudar para o modo de baixo consumo de energia para o consumo até que receba um sinal de partida de MCU novamente [10].

3.5.2 Sensor PIR

O Sensor PIR (*Passive Infra-Red*) mostrado na figura 3.5.2.1 é um dispositivo piroelétrico que detecta movimento através das alterações nos níveis de infravermelho (calor radiante) emitidos por objetos circundantes.

Este movimento pode ser detectado através da verificação de uma mudança brusca no padrão IR (Infravermelho) no ambiente.

Quando é detectado movimento o sensor emite um sinal de PIR alto no seu pino de saída. Este sinal lógico pode ser lido por um micro controlador ou utilizado para acionar uma carga externa.

Dispositivos Piroelétricos, tais como o sensor PIR, possuem elementos feitos de um material cristalino que gera uma carga elétrica quando expostos a energia de raios infravermelhos.

O dispositivo contém uma lente de Fresnel, que concentra os sinais infravermelhos sobre o elemento. Como os sinais infravermelhos ambientais mudam rapidamente, coloca-se um amplificador embarcado na saída para indicar movimento.



Figura 3.5.2.1 - Sensor PIR [11]

Alimentação e Ligação

O sensor PIR funciona com tensões na faixa de 3 a 6V DC com 130 μ A em modo de aguardo e 3mA quando ativo, sendo sua temperatura de operação de 0°C a 50°C. A ligação dos pinos do sensor pode ser observada na figura 3.5.2.2.

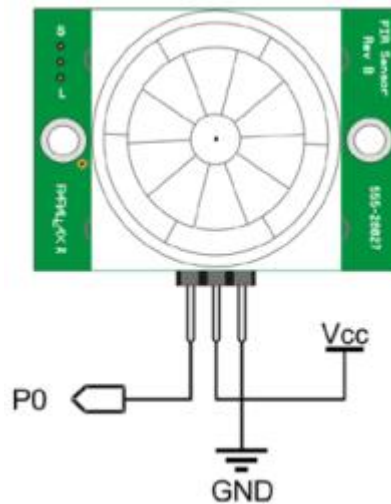


Figura 3.5.2.2- Ligação dos pinos [11]

A sua comunicação do tipo *Single bit* podendo ser configurado como ativo alto ou baixo.

Seu alcance é afetado por:

- As propriedades de tamanho térmicas de objetos próximos
- Condições ambientais, incluindo a temperatura ambiente e fontes de luz.

O gráfico na figura 3.5.2.3 mostra os efeitos aproximados de temperaturas conhecidas na área de detecção do sensor de PIR de um adulto. Este dispositivo foi projetado para uso indoor. Operação no exterior ou em temperaturas extremas pode afetar negativamente a estabilidade.

A exposição direta à luz solar ou outras formas de aquecimento radiante possa causar operação indesejada.

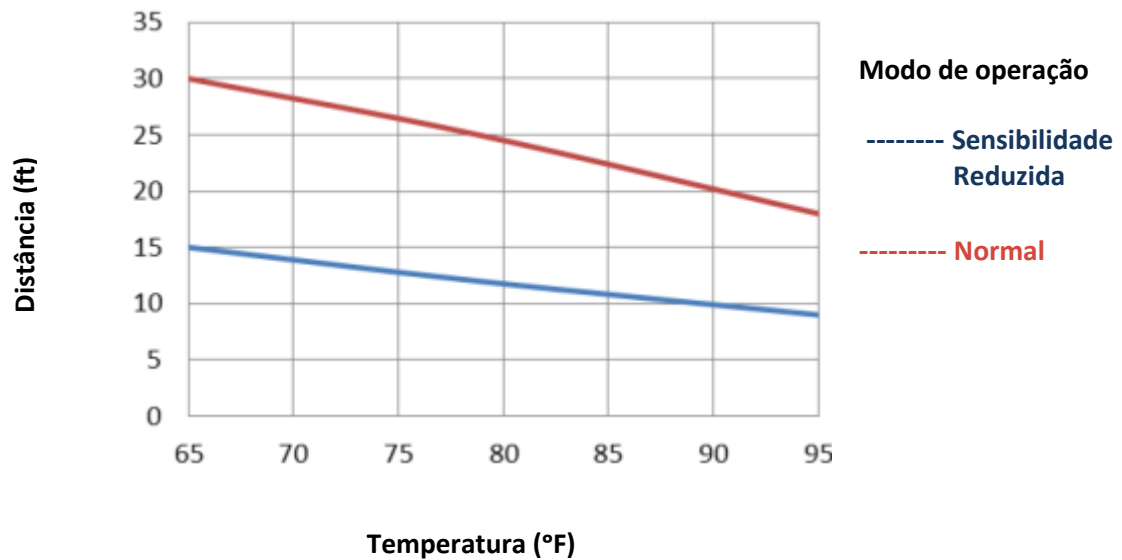


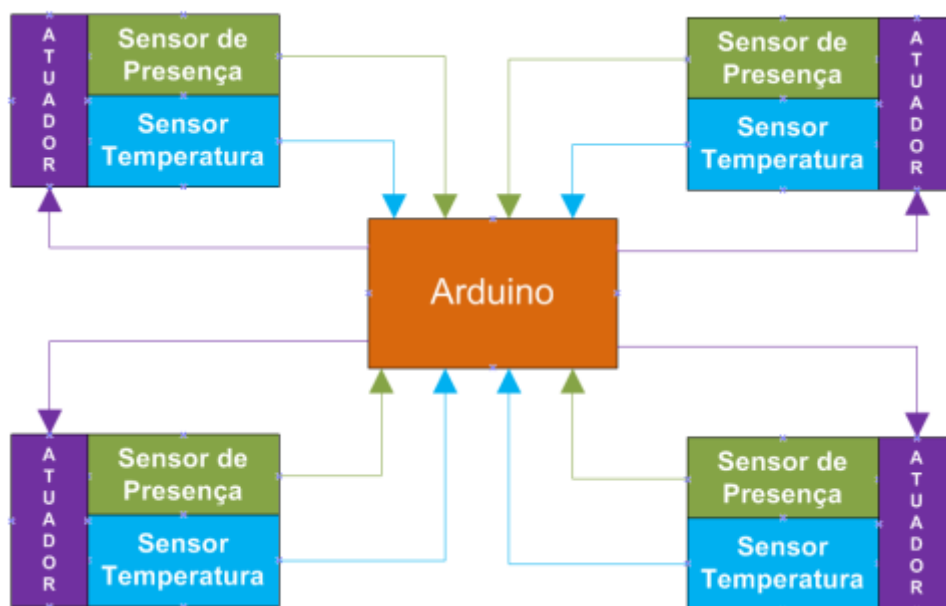
Figura 3.5.2.3 - Efeito da temperatura na distância de detecção do Sensor PIR [11]

Considerando a temperatura ótima 23°C a 27°C (73°F a 80°F) o sensor funciona de forma efetiva, considerando os ambientes de uma casa padrão.

4. Metodologia

O sistema proposto no projeto consiste em criar quatro ambientes diferentes os quais vão contar com um sensor de temperatura e umidade em cada ambiente e um controlador Arduino para controlar todos os sensores dos ambientes. Através desse controle será colocado um atuador a ser escolhido que fará o controle da temperatura do ambiente.

Os sensores e atuadores serão ligados ao modulo Arduino e através do GUI será possível fazer o monitoramento do sistema através do modulo ethernet do Arduino o diagrama de blocos da figura 4.1, mostra como será feita a ligação dos componentes do sistema.



Figura

4.1- Diagrama de blocos do projeto

4.2 Materiais utilizados

Nesse projeto serão utilizados os seguintes materiais:

- Quatro Sensores de presença PIR
- Quatro Sensores de Temperatura e umidade DTH11
- Arduino Mega
- *Ethernet Shield*
- *Protoboard*
- Resistores, Capacitores e Transistores diversos.
- Fios de conexão
- Fonte de Alimentação 12 v
- Cabo USB

4.3 Conversões do Sinal

Observando o *Datasheet* do micro controlador ATmega48PA pode-se observar as seguintes características elétricas. [12]

Temperatura de Operação	-55 °C a 125°C
Temperatura de armazenamento	-65 °C a 150°C
Voltagem em qualquer pino exceto RESET com a relação ao GND	-0.5V até Vcc + 0.5V
Voltagem no RESET com relação ao <i>Ground</i>	Até 13V
Voltagem máxima de operação	6V
Corrente DC por pino I/O	40mA
Corrente DC no pino Vcc e GND	200mA

Desta forma pode-se notar que a corrente máxima do Arduino é de 40mA por entrada / saída em cada pino. Para evitar danificar o Arduino deve-se utilizar um Driver, desta forma vamos garantir que o Arduino use apenas a corrente que ele consegue garantir de maneira segura por porta.

Uma forma de se criar esse Driver seria utilizando-se de um transistor como uma chave eletrônica, um transistor pode operar como chave eletrônica, bastando para tal polarizá-lo de forma conveniente: corte ou saturação.

Quando um transistor está saturado opera como um curto (chave fechada) entre o coletor e o emissor de forma que VCE aproximadamente igual a 0 v e quando está no corte, opera como um circuito aberto (chave aberta) entre o coletor e o emissor, de forma que VCE aproximadamente igual a VCC.

No chaveamento eletrônico com transistores, devemos levar em conta dois tipos de saturação: fraca e forte.

Na saturação fraca, a corrente de base é suficiente para levar o transistor à saturação. Tal procedimento, porém não é aconselhável visto que pode haver uma variação de VCC e na própria corrente de base de saturação (IB SAT).

Utiliza-se normalmente a saturação forte, que assegura a condição de saturação para todos os valores de VCC. Uma regra prática é considerar a corrente de base como 1/10 da corrente de saturação de coletor.

Controle PWM

O controle proporcional, conforme descrito na secção 2.6, que pode ser realizado pelo Arduino é feito por um método chamado Modulação por Largura de Pulso, mais conhecida pela sigla inglesa PWM (*Pulse-Width Modulation*), é uma técnica largamente utilizada para variar a potência entregue a uma carga.

O Arduino utiliza o PWM para simular sinais analógicos a partir de saídas digitais. O PWM consiste numa onda quadrada, chaveando a saída entre ligado e desligado, conforme indicado na figura 4.3.1. O controlador utiliza 5 v para ligado e 0 v para desligado.

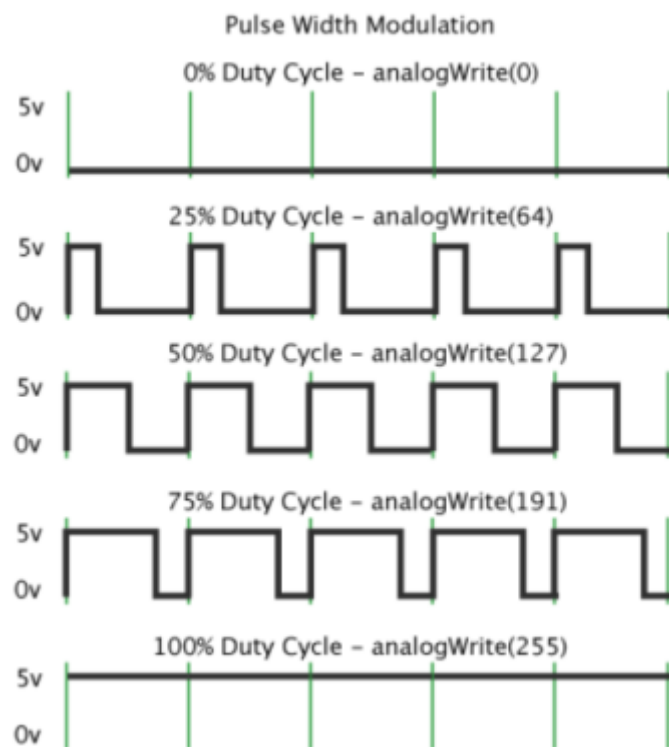


Figura 4.3.1- PWM do Arduino [8]

4.4 Arquitetura do Sistema

Através da Linguagem de Programação foram montados vários códigos que fazem a interação dos sinais recebidos pelo Arduino e o usuário.

Este projeto conta com a criação de uma interface gráfica para visualizar um histórico de temperaturas e umidades ao longo de um período e também conta com a possibilidade de configurar a temperatura de set point do sistema.

Para dar início a esse sistema, foi necessária a escolha de um sistema de gerenciamento de banco de dados e uma linguagem, no caso, escolhidos o MySQL e o PHP respectivamente, ambos rodando em um servidor Apache sustentado pelo aplicativo Wamp.

O banco de dados conta com uma estrutura que armazena em uma tabela os ambientes disponíveis, 04 no caso e em outra tabela, os valores de temperatura e umidade. Essa escolha de estrutura facilita a inserção de um novo ambiente ao software caso necessário.

Como o software aplicativo não estará em operação contínua necessariamente, tal como em momentos em que não haverá ninguém em nenhum dos ambientes, optou-se por criar um código que faz a leitura dos valores, vindos do servidor levantado pelo Arduino, e os armazena nas tabelas. Para que a operação seja realizada no servidor automaticamente, fez-se uso do aplicativo php.exe, disponível no próprio servidor Apache, para executar esse código de web e o mesmo foi agendado através de uma *shell* do Windows para rodar na periodicidade escolhida. Nessa forma, a escolha de periodicidade dos gráficos apresentados se dará apenas pela escolha de periodicidade no agendamento da tarefa.

Uma vez que os dados são salvos, a aplicação web tem a função de realizar a leitura no banco de dados e repassar para a visualização. Foi realizado o uso de *JQuery* para auxiliar na leitura assíncrona do banco de dados. Dessa maneira, a página web não precisa ser atualizada para que os dados sejam atualizados na tela, retirando a usabilidade incômoda de a aplicação piscar a tela. O *JQuery* também auxilia no desenvolvimento de animações melhorando a interface gráfica do sistema.

A interface através do *JQuery* e dessa leitura assíncrona no banco de dados, atualiza os valores de temperatura e umidade nos quadros que representam os ambientes. O mesmo mecanismo funciona para a atualização do gráfico. O mesmo busca os últimos valores gravados na *shell* do Windows e dispõe os valores. O gráfico é montado por um plug-in em Java script que necessita na entrada, de uma matriz dos valores x,y e o mesmo identificam os valores máximo e mínimo no eixo y e divide o eixo na melhor forma possível.

A planta baixa da casa foi feita com um efeito que simula a luminosidade em um ambiente. Esse efeito foi realizado criando quadros (divs) com uma opacidade alta de modo a não atrapalhar a imagem de fundo. Com isso, bastou mudar a visualização “*on – off*” das imagens de acordo com os dados vindos do banco.

A temperatura de set point foi programada com o evento de mudança de valor do *slider* presente no sistema.

Quando o usuário troca essa temperatura, o site faz uma requisição ao servidor levantado pelo Arduino enviando o valor do mesmo como parâmetro para o Arduino.

Desta forma foi construída a seguinte tela para a Interface gráfica com usuário, mostrado na figura 4.4.1

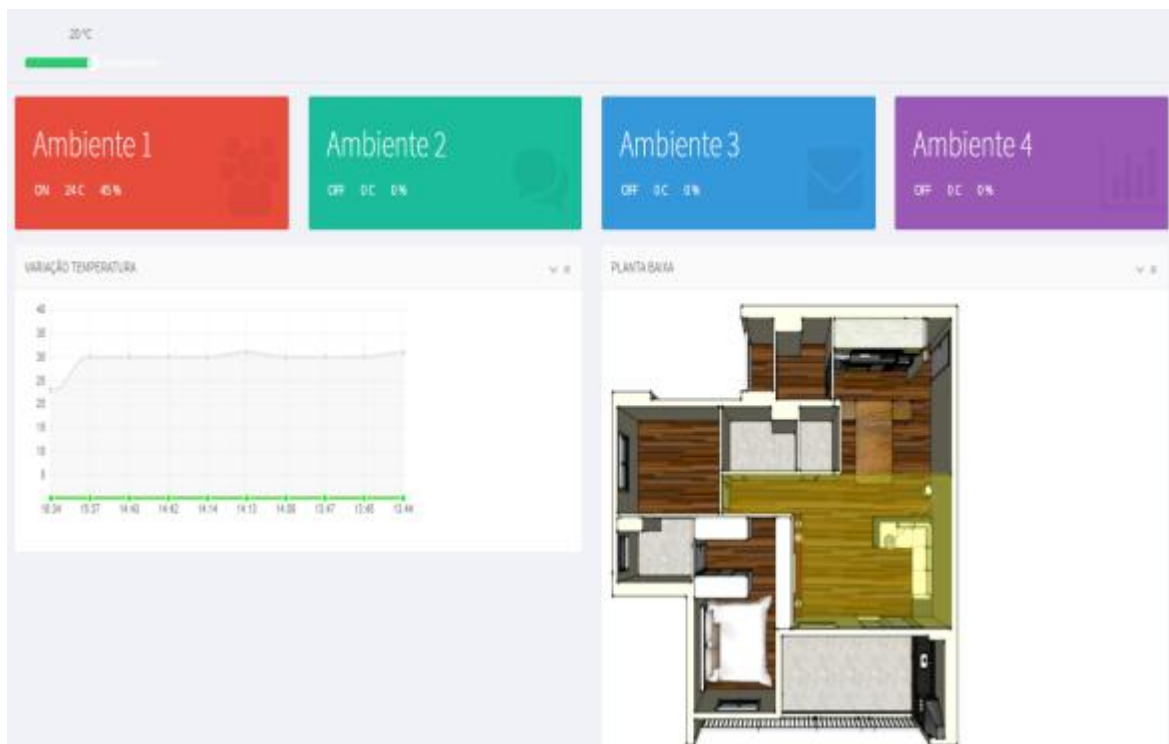


Figura 4.4.1: Tela principal do sistema

Os códigos descritos abaixo foram utilizados para se programar o a interface gráfica e coletar os valores dos sensores coletados pelo Arduino, as linhas de programação são mostradas na parte de anexos desse relatório.

Salvar.php:

Código responsável por inserir os valores vindos do Arduino para o banco de dados. O mesmo acessa o IP Fixo disponibilizado no Arduino, faz a leitura do conteúdo, divide a string em um vetor e salva os valores no *MySql*. Este é o código executado pelo php.exe e agendado na *shell* do Windows.

Index.js:

Esse é o código Java script responsável por controlar a página do aplicativo. O mesmo faz os acessos aos arquivos PHP de forma assíncrona através do *JQuery* e passa essas informações ao aplicativo web. Nesse código, há uma programação destinada a atualizar os quadros com a temperatura, umidade e presença de pessoas no ambiente, enviar a temperatura de set point ao Arduino, controlar os quadros que dão o efeito de luminosidade na planta baixa, fazer a requisição da atualização do gráfico e enviar ao Arduino a necessidade de ligar o ar em cada um dos ambientes.

DelayLuz.php:

Este código tem como objetivo controlar o horário em que uma pessoa entra em algum ambiente. Isso deu a possibilidade de aguardar 60 segundos de inatividade no ambiente antes que o sistema seja desligado. Com isso, o sistema não irá ficar ligando e desligando a cada presença ou não presença de alguma pessoa.

DadosTemperatura.php:

Este código tem como objetivo buscar no banco de dados, os últimos valores de temperatura e seus horários para alimentar o gráfico do aplicativo. Conforme mostrado na figura 4.4.2.

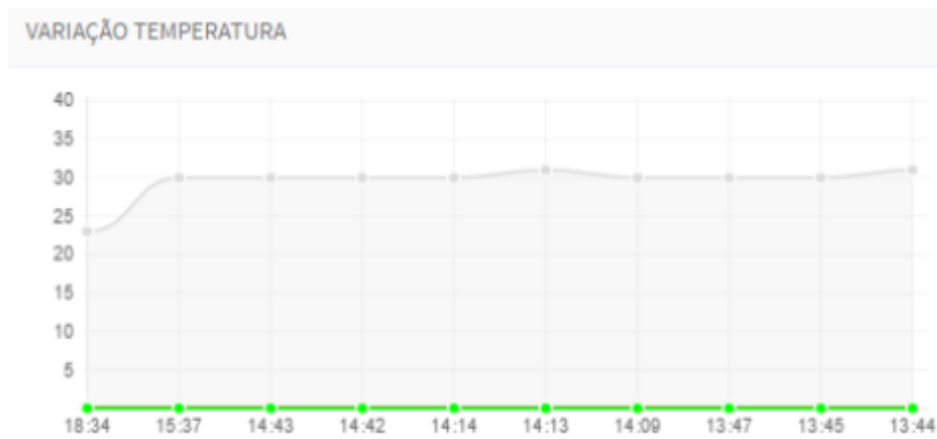


Figura 4.4.2: Gráfico gerado com valores coletados dos sensores

4.5 Diagrama de blocos

Os seguintes diagramas de blocos mostram detalhadamente o funcionamento do software

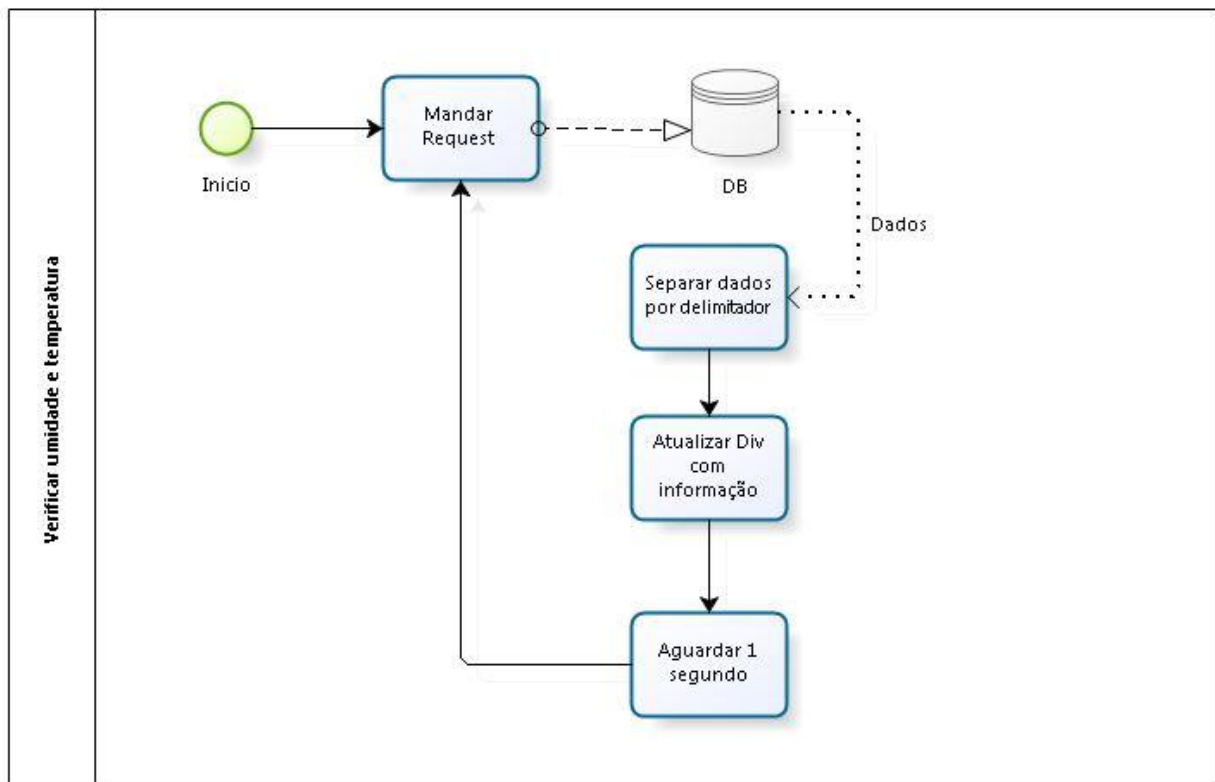


Figura 4.5.1: Diagrama de Blocos de verificação de temperatura e umidade

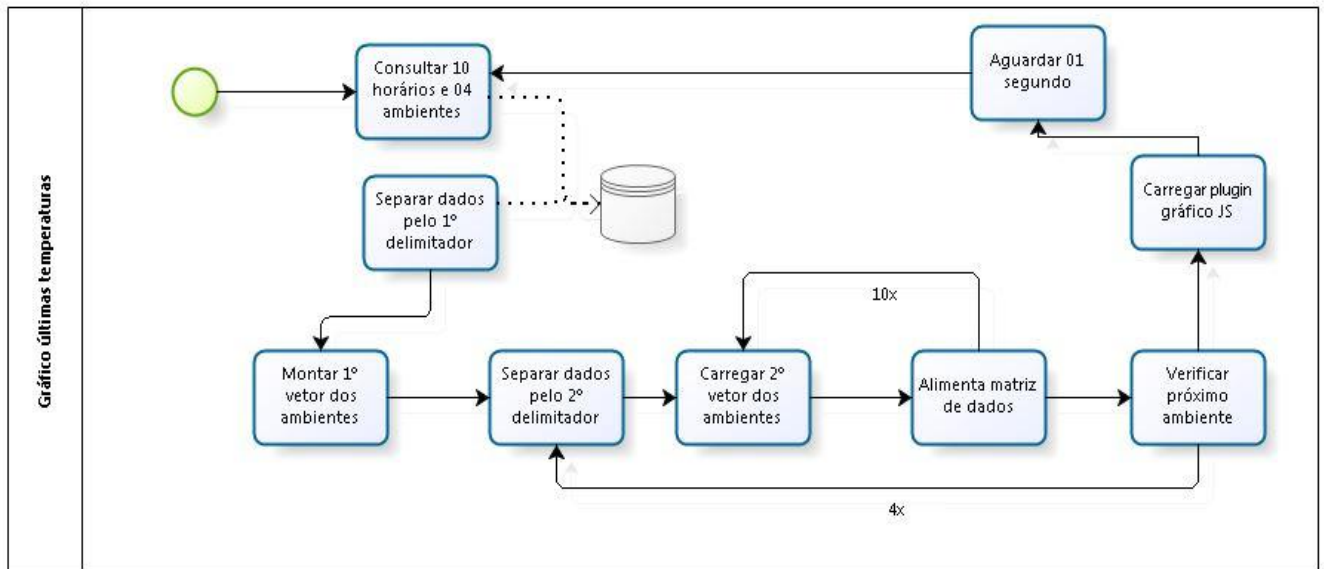


Figura 4.5.2: Diagrama de Blocos para gravação da ultima temperaturas

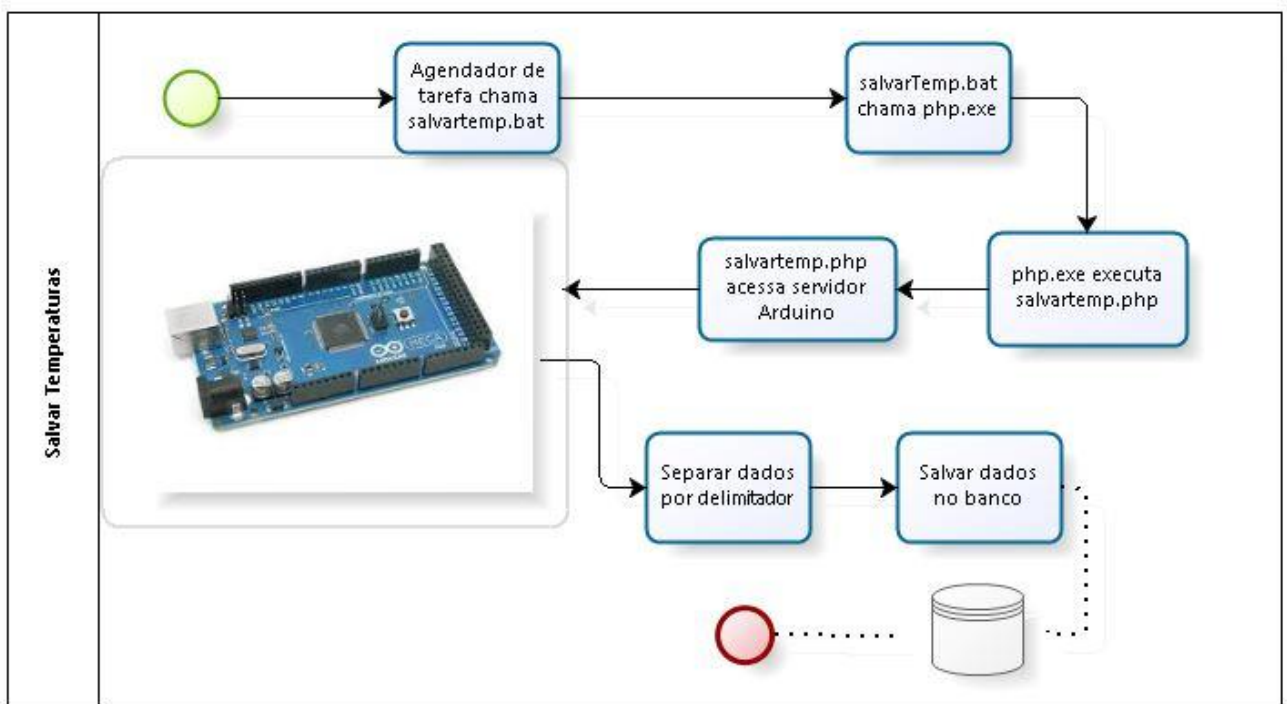


Figura 4.5.3: Diagrama de Blocos para salvar a temperatura

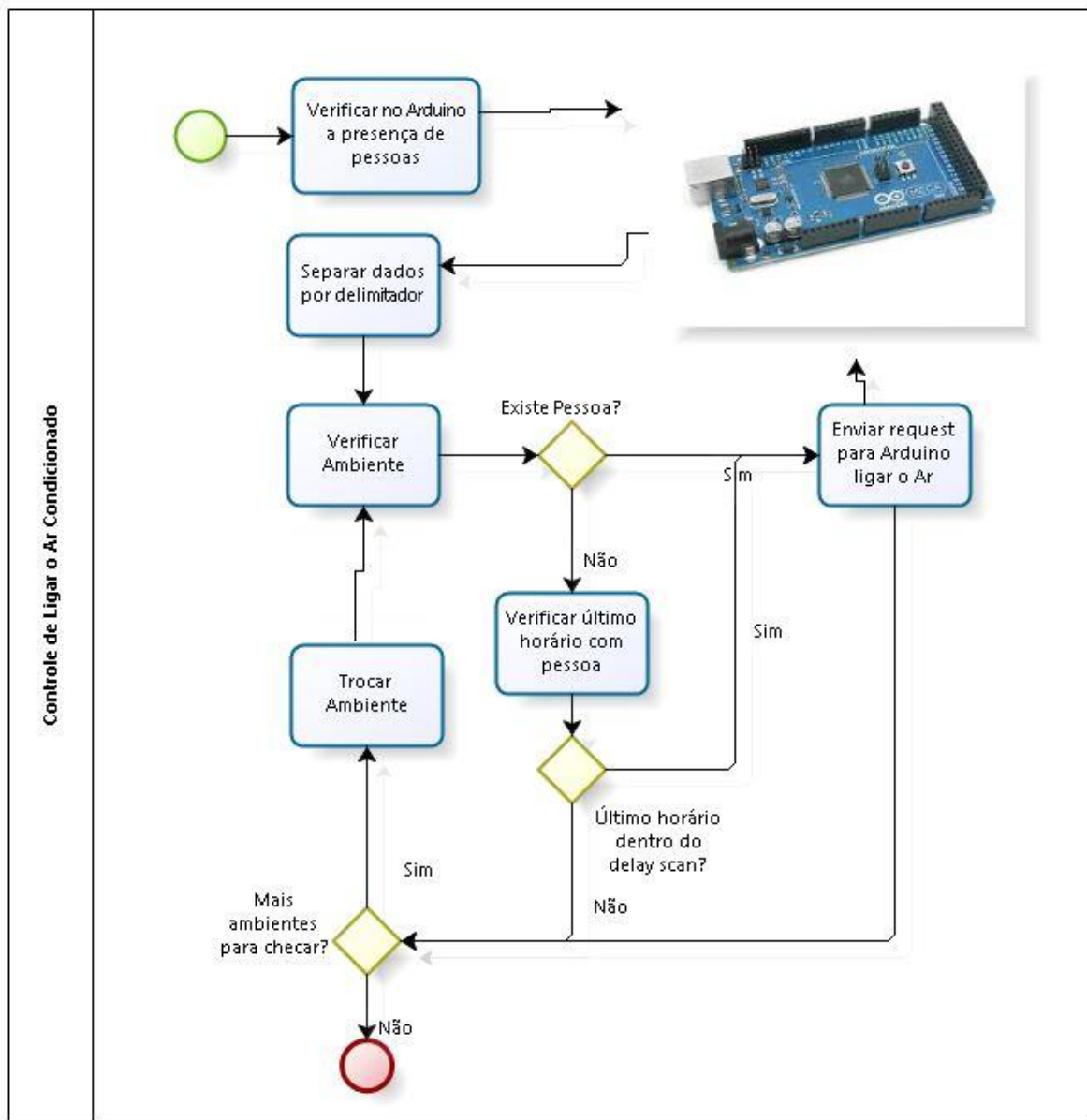


Figura 4.5.4: Diagrama de Blocos de controle de ligar e desligar o sistema

5. Resultados

Nesta seção será apresentado o projeto desenvolvido bem como todas as integrações realizadas nos dispositivos. As principais características de desempenho do sistema serão exibidas e executados no software web que foi desenvolvido, através dos dados adquiridos dos sensores conectados ao Arduino em tempo real, podendo ser observados na tela e armazenados no banco de dados.

A proposta do sistema é monitorar sensores e gerar comandos para atuadores por meio de ação requisitada por um usuário operador ou de forma automática, realizada em configuração programada no software web.

5.2 Funcionamento do Sistema

O sistema proposto consiste de quatro sensores de temperatura e umidade e quatro sensores de presença e foram configuradas quatro saídas para ativar quatro atuadores de saída digital, que podem enviar comandos para ligar ou desligar equipamentos, conectados nessas saídas, conforme mostrado na figura 5.2.1

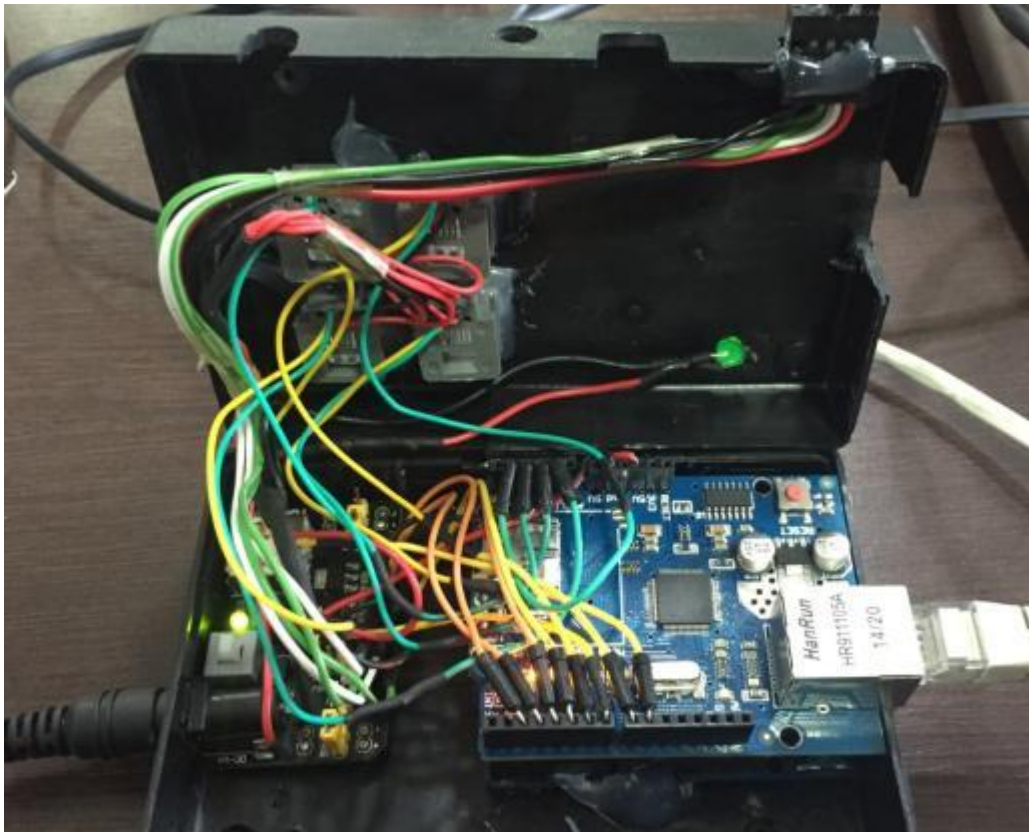


Figura 5.2.1: Ligação dos sensores e saídas para atuadores na placa do arduino

Os sensores foram montados em uma caixa de plástico e conectados a um cabo de quatro vias, as quais correspondem ao VCC, Terra, Sinal do sensor de presença e sinal do sensor de temperatura e umidade conforme mostrado na figura 5.2.2 e montado na sua caixa conforme figura 5.2.3:



Figura5.2.2: ligação dos sensores no cabo



Figura 5.2.3: Sensores montados na caixa

O sistema funciona através do software web que deve ser montado em um computador servidor que fará a comunicação com o Arduino através da rede local TCP-IP. O Arduino ao receber os comandos vindos do software web, traduz a informação dos comandos e executa as ações necessárias em cada uma das quatro saídas para os atuadores. O fluxo da figura 5.2.4 ilustra como funcionam os comandos.

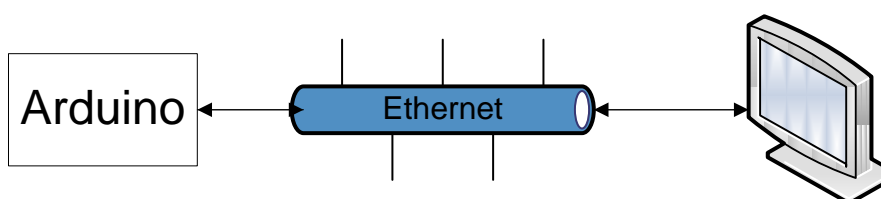


Figura 5.2.4: Funcionamento dos comandos

O arduino manda as informações de temperatura, umidade e presença em tempo real e o sistema faz o controle de ações das saídas para os atuadores em ciclos de tempo conforme o código *delay* descrito na seção 4.4

Todo o controle nos atuadores funciona de forma automática sem a participação do usuário, através dos comandos pré-estabelecido pelo usuário, como temperatura de *set point* e tempo de *delay*.

5.3 Sensores e atuadores

As informações enviadas pelos sensores podem ser observadas pelo navegador de internet através do endereço 192.168.1.102 onde são mostrados todos os dados coletados pelos sensores *como* uma lista de pontos (variáveis). A figura 5.3.1 mostra a tela dos dados dos sensores.

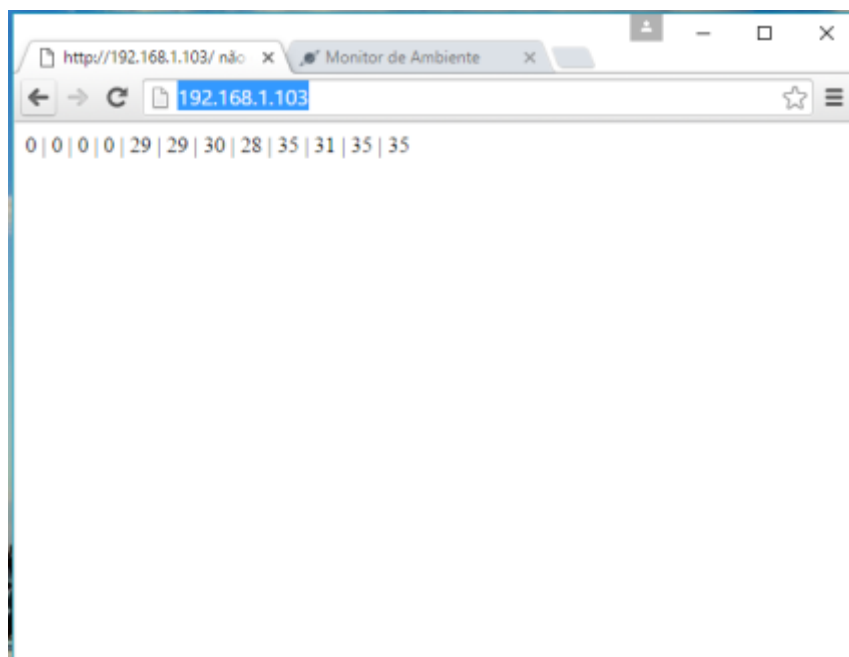


Figura 5.3.1: dados enviados dos sensores

5.4 Cenário

O ambiente gráfico que será utilizado pelos usuários do sistema é simbolizado em uma tela gráfica que possui a planta baixa do local o ajuste de temperatura de *set point* um mostrador individual de cada ambiente onde se pode ver a temperatura, a umidade, e a presença de habitante foi detectada e um gráfico registrando a variação de temperatura nos quatro ambientes configurados conforme mostrado na figura 4.4.1 onde a arquitetura do sistema pode ser observada na seção 4.4.

Na tela principal é configurada a temperatura de *set point* através da seguinte barra mostrada na figura 5.4.1

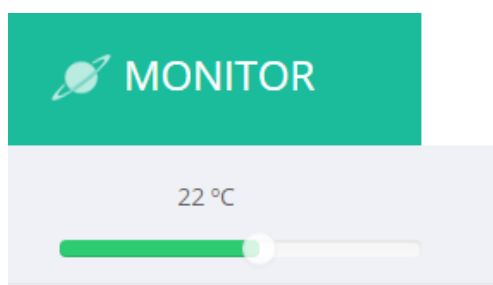


Figura 5.4.1: Barra de ajuste da temperatura de Set Point (Temperatura Ideal).

Em um sistema de refrigeração, por exemplo, só haverá alguma atividade nos atuadores se a temperatura medida pelos sensores estiver acima da temperatura de *Set point*, esta será uma das condições para testes do sistema.

Na tela principal também se observam as condições de temperatura e umidade de cada ambiente, conforme figura 5.4.2, onde é possível se observar respectivamente se existe presença de habitante (ON, se existe habitantes, OFF caso contrário) bem como a temperatura e umidade local



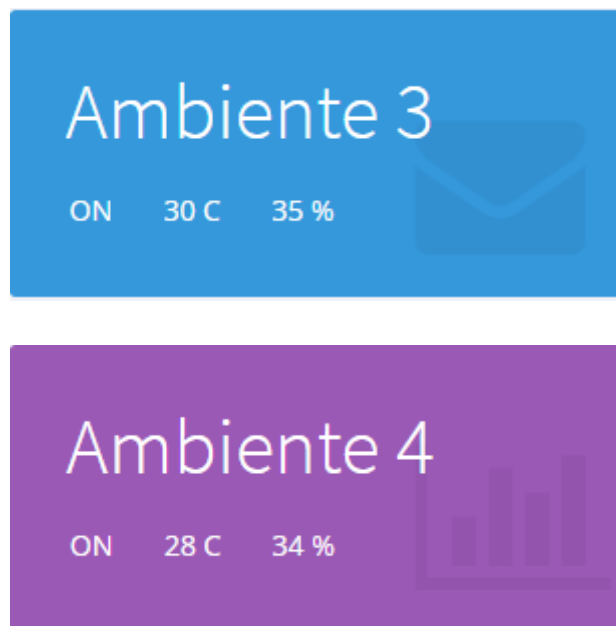


Figura 5.4.2: Dados dos sensores para cada ambiente

Para indicar a presença de habitantes foi inserida a planta baixa no sistema e quando a presença é detectada no sistema o local ficara iluminado em amarelo, conforme mostrado pela figura 5.4.3:



Figura 5.4.3: Planta baixa do sistema, onde se pode observar a presença detectada em três dos quatro ambientes monitorados.

Para monitorar e mostrar as variações de temperatura foi montando um gráfico com escala em Y variável para mostrar a temperatura combinada dos quatro ambientes, através da figura 5.4.4 se pode ver o gráfico onde cada cor de linha corresponde ao ambiente conforme mostrado na figura 5.4.2, o intervalo de aquisição dos dados foi estipulado em 5 minutos.

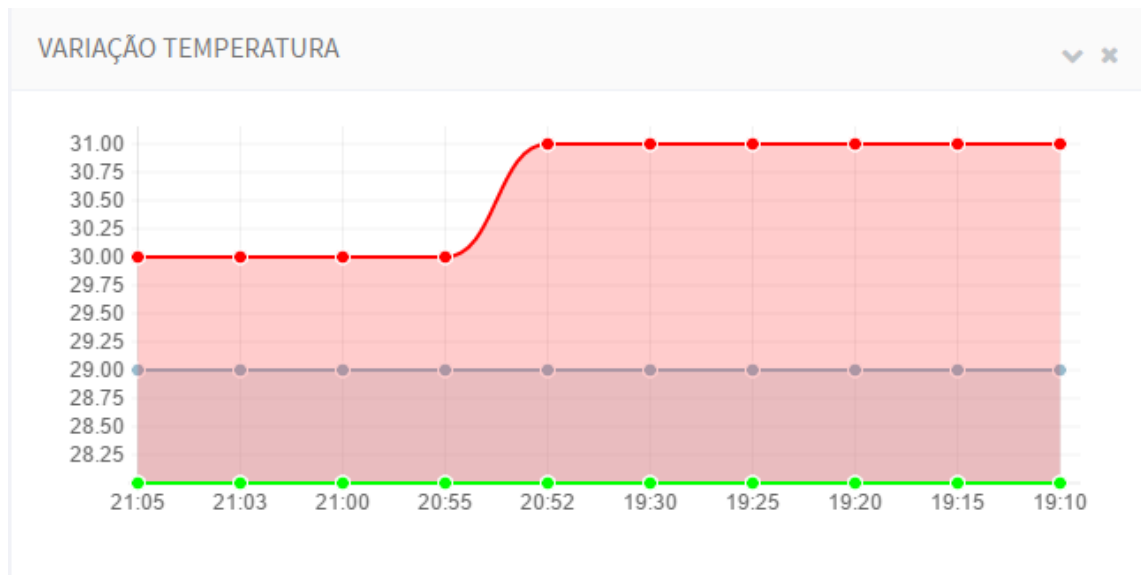


Figura 5.4.4: Variação da temperatura em relação ao tempo de aquisição

5.5 Análises de Resultados

O trabalho propôs a montagem de uma rede de sensores e saídas atuadores utilizando, o Arduino redes de computadores e programação Web, usual em ambientes residenciais na função de supervisor.

Para a análise de resultados, foi proposta uma simulação, onde foi configurada uma temperatura de set point de 30° C e foi testado com o sistema respondia com relação aos dados coletados e as saídas nos atuadores, como saída foram usadas um LED para indicar se a saída estava ligada ou desligada.

No primeiro caso observou-se que a temperatura de set point estava configurada para 30° C e que a temperatura medida no ambiente 1 era de 28 °C nesse caso com era esperado o LED referente ao atuador do Ambiente 1, considerando que o sensor detectou presença no ambiente analisado, deveria estar desligado as figuras 5.5.1 e 5.5.2 indicam que o sistema respondeu corretamente ao do teste

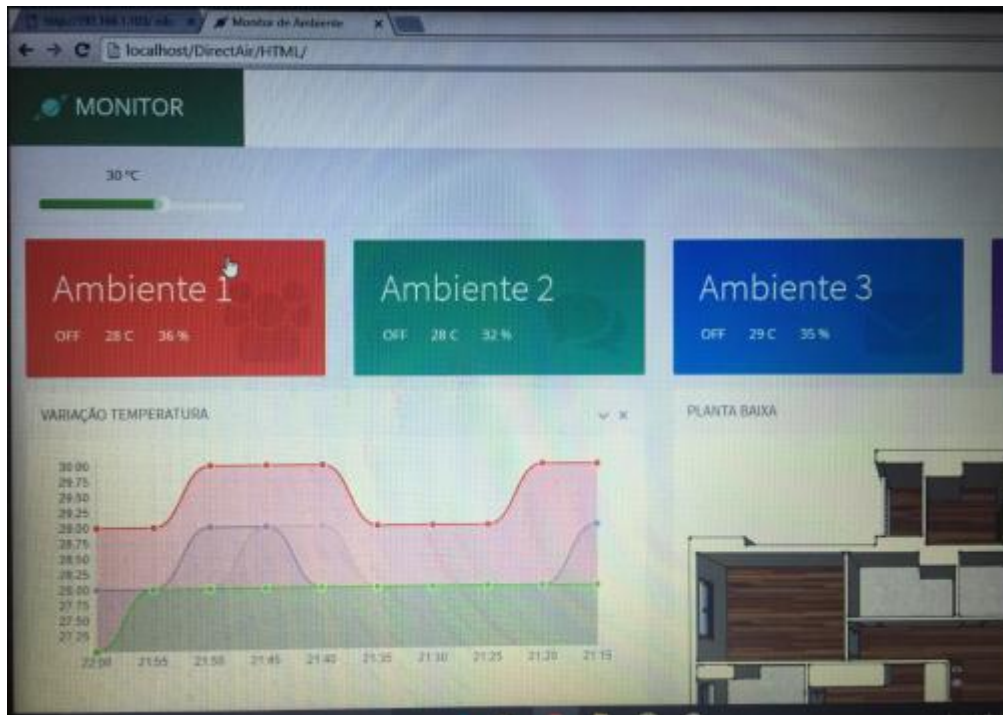


Figura 5.5.1: Monitor do ambiente indicando temperatura de *Set Point* em 30° C e mostrando a temperatura do ambiente 1 em 28°C

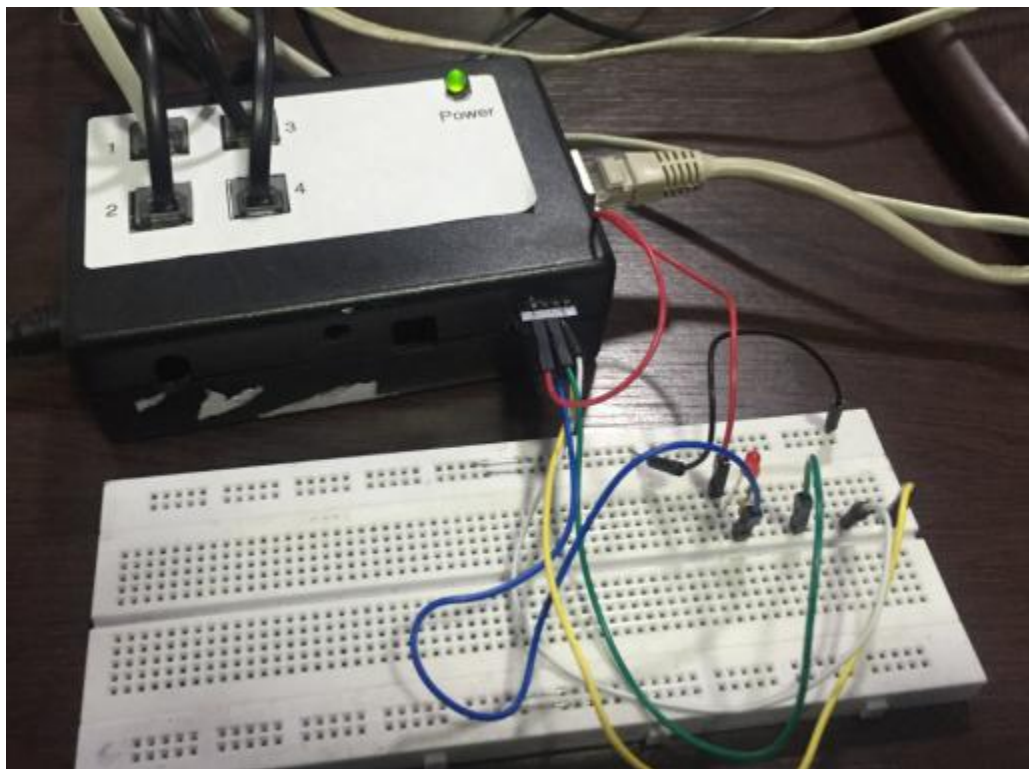


Figura 5.5.2: quatro saídas do atuador indicando LED apagado, atuador desligado

No segundo caso observou-se que a temperatura de *set point* estava configurado para 18° C e que a temperatura medida no ambiente 1 era de 28 °C nesse caso com

era esperado o LED referente ao atuador do Ambiente 1 deveria estar ligado, considerando que o sensor detectou presença no ambiente analisado, as figuras 5.5.3 e 5.5.4 indicam que o sistema respondeu corretamente ao do teste.

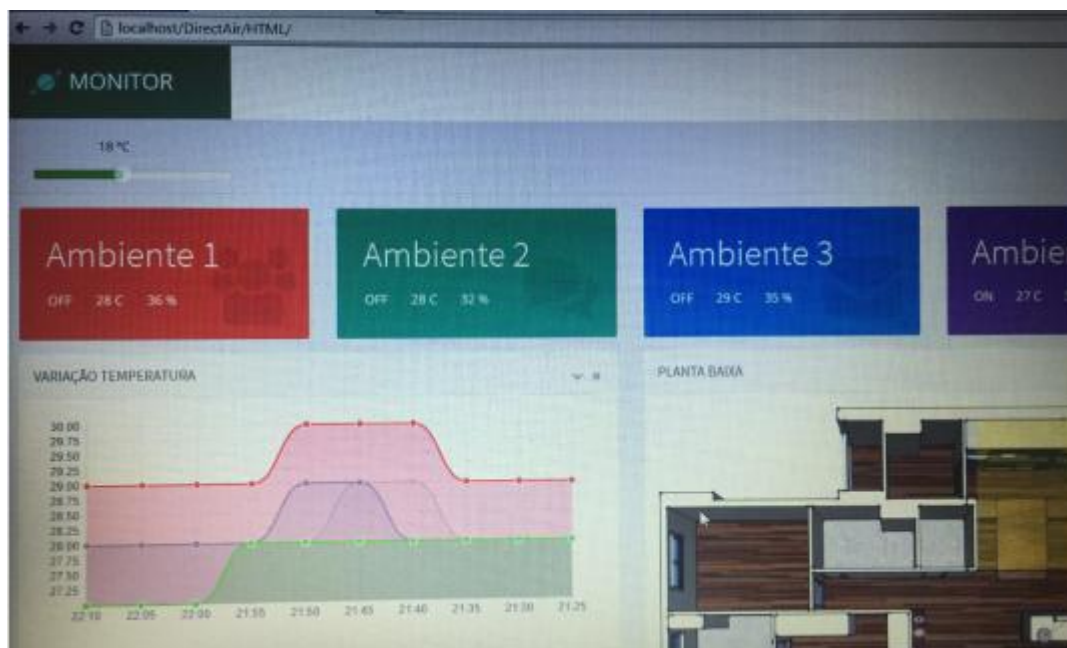


Figura 5.5.3: Monitor do ambiente indicando temperatura de *Set Point* em 18° C e mostrando a temperatura do ambiente 1 em 28°C

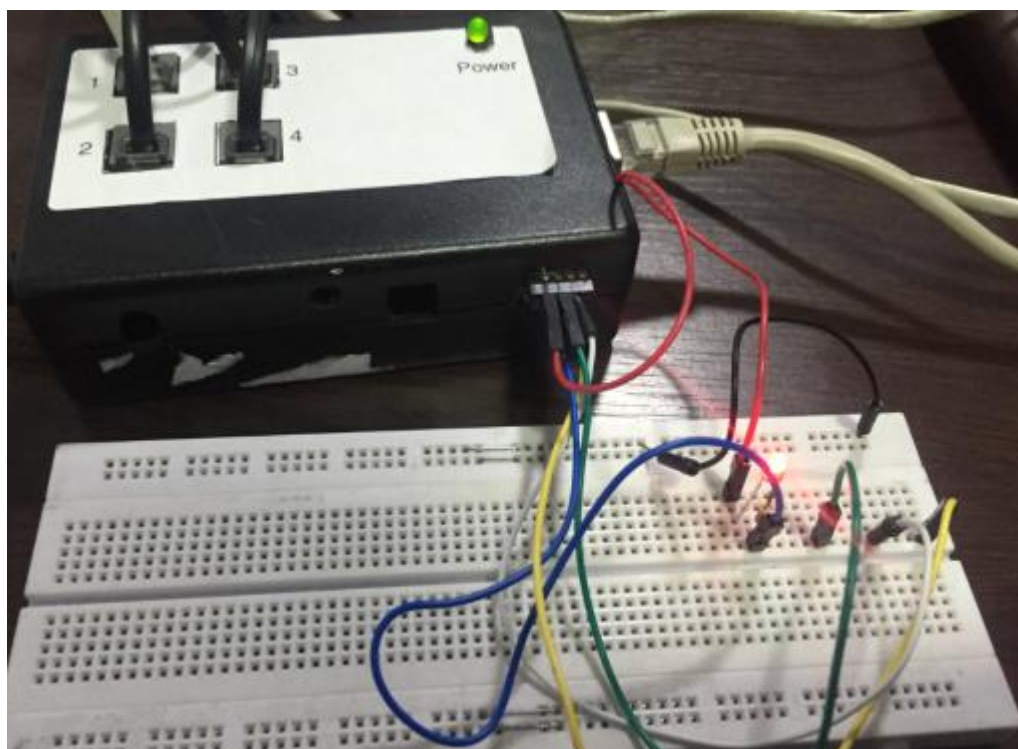


Figura 5.5.4: quatro saídas do atuador indicando LED aceso, atuador ligado.

Em um terceiro teste observou-se que para qualquer temperatura de *set point*, sem detecção de habitantes na residência neste caso não havia nenhuma saída nos atuadores indicando que o sistema respondeu corretamente aos dos testes

Observa-se que o software desenvolvido em programação WEB cumpriu seu papel por suas capacidades de monitoração de processo, desenvolvimento de interface gráfica, armazenamento de dados e integração com os dispositivos das Redes de computadores.

6. Conclusões

Neste trabalho foi possível alcançar os objetivos sendo possível monitorar a temperatura e umidade dos quatro ambientes propostos na casa. Com esse monitoramento foi possível realizar o acionamento de outros dispositivos respeitando as configurações determinadas pelo sistema, como a temperatura de set point para acionamento de climatização.

Foi possível montar com sucesso uma interface gráfica onde o usuário pode monitorar o sistema e definir a temperatura de set point e o tempo de escaneamento do sistema, uma vez que o usuário está no local, pode-se determinar quanto tempo depois o sistema fara um novo escaneamento, por todo esse intervalo de tempo o sistema vai manter as configurações determinadas e não fará alteração na climatização.

Por se tratar de um sistema baseado em protocolo TCP-IP a expansão do sistema é possível através de redes locais, para o caso de ambientes muito distantes pode-se conectar outro Arduino e este se conectar ao servidor desta forma receber e enviar dados aos sensores e atuadores no ambiente que for instalado, no caso é possível se utilizar de uma rede WI-FI, utilizando um shield WI-FI conectado ao Arduino, desta maneira se utilizaria cabos com comprimento menor para ligar os sensores reduzindo a perda de informações nos cabos.

O desenvolvimento deste sistema pode ser utilizado para criação de novos sistemas de climatização residencial e pode auxiliar no consumo sustentável de energia para a climatização de ambientes uma vez que se pode instalar sensores que possam medir em tempo real o consumo de energia de uma climatizador, reduzindo o gasto com climatização de ambientes vazios e variando a temperatura gradativamente de acordo com a temperatura de set point.

O sistema pode verificar a umidade do ar e de acordo com o valor considerado mínimo o sistema pode não ativar o sistema de ar condicionado e dar preferência para um sistema de climatização através de um umidificador ou até mesmo combinando os dois sistemas obtendo-se uma melhor qualidade no ar e uma temperatura confortável para os habitantes.

Em suma o trabalho apresentou êxito no seu protótipo e este abre diversas possibilidades de expansão e melhoria, uma vez que grande parte da inteligência do sistema está no código que roda em um computador a capacidade de expansão é muito superior à capacidade de se aplicar o sistema apenas no Arduino que apresenta limitações de processamento e capacidade armazenamento desta forma o Arduino funciona como uma interface entre computador e dispositivos, como sensores e atuadores, sendo esse de fácil aplicação e baixo custo de montagem, e utilização de técnicas de programação, eletrônica e conhecimentos de redes de computadores.

7. Referências Bibliográficas.

- [1] VILLANI , Emilia; Abordagem Híbrida para modelagem de Sistemas de Ar Condicionado em edifícios inteligentes E.Villani - São Paulo 2000
- [2] JABARDO, J. M. S (1984) Conforto Térmico das Habitações; Eternit/IPT, São Paulo
- [3] ASHRAE, 2004: ASHRAE Standard 55-2004, Thermal environmental conditions for human occupancy. American Society of Heating, Refrigerating and Air-conditioning Engineers, Inc., Atlanta.
- [4] SUZUKI, Eliane Hayashi, escritórios com climatização artificial na cidade de São Paulo/ E.H Suzuki – São Paulo 2010
- [5] S. Lee et al.: A Pyroelectric Infrared Sensor-based Indoor Location-Aware System for the Smart Home
- [6] Ogata, Katsuhiko.: Engenharia de Controle Moderno 5ª edição – São Paulo: Pearson Prentice Hall, 2010
- [7] McRoberts, Michael Arduino Básico/Michael McRoberts [tradução Rafael Zanolli]. —São Paulo: Novatec Editora, 2011.
- [8] <https://www.arduino.cc/en/Main/arduinoBoardMega>, acessado em 20/11/2015
- [9] Greg Wilson (2006). "Off With Their HUDs!: Rethinking the Heads-Up Display in Console Game Design". Gamasutra. Retrieved February 14, 2006.
- [10] DHT11 Humidity & Temperature Sensor D-Robotics UK www.droboticsonline.com
- [11] PIR Sensor Rev B Product Guide - www.parallax.com/product/555-28027
- [12] 8-bit Atmel ATmega48PA Microcontroller

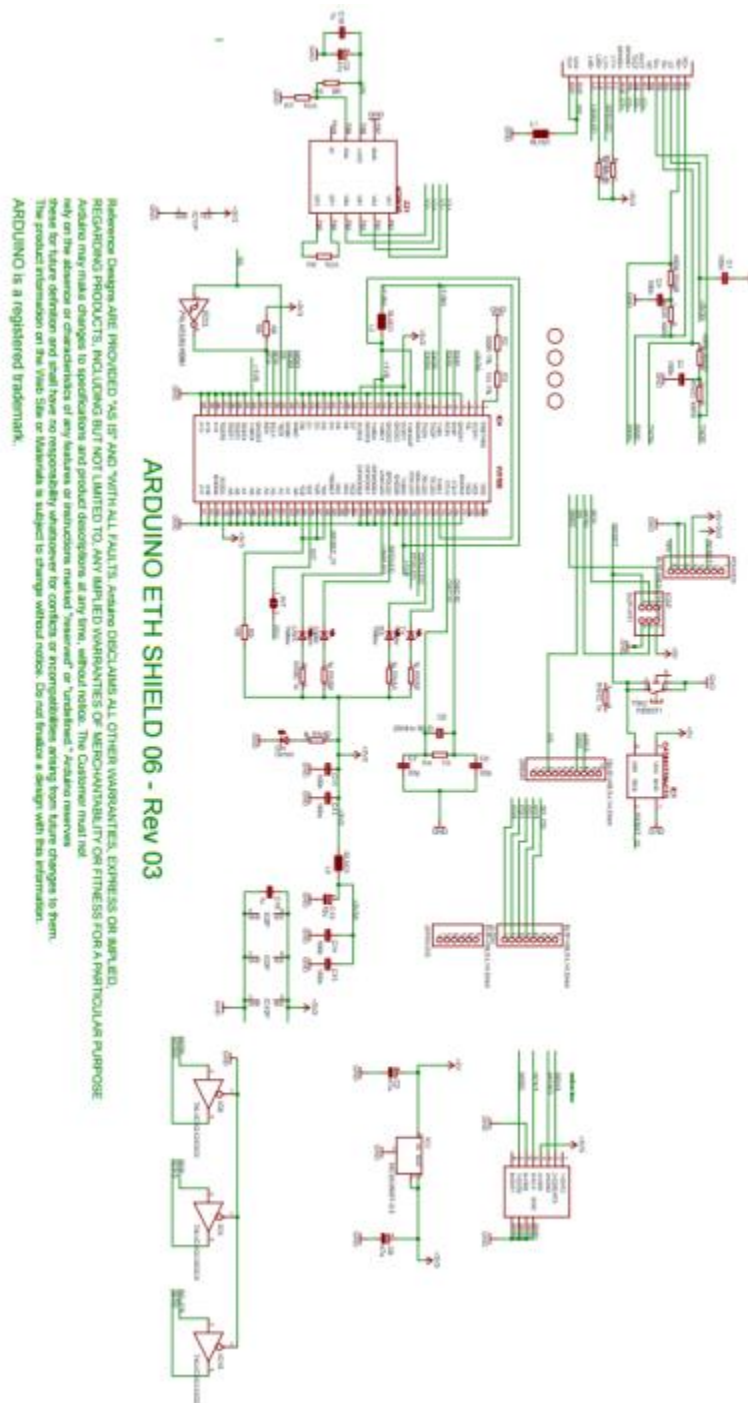
<http://www.atmel.com/Images/doc8161.pdf>

[13] MIYAGI, P.E. Controle Programável – Fundamentos do Controle de Sistemas a Eventos Discretos, Editora E. Blucher, São Paulo, 1996

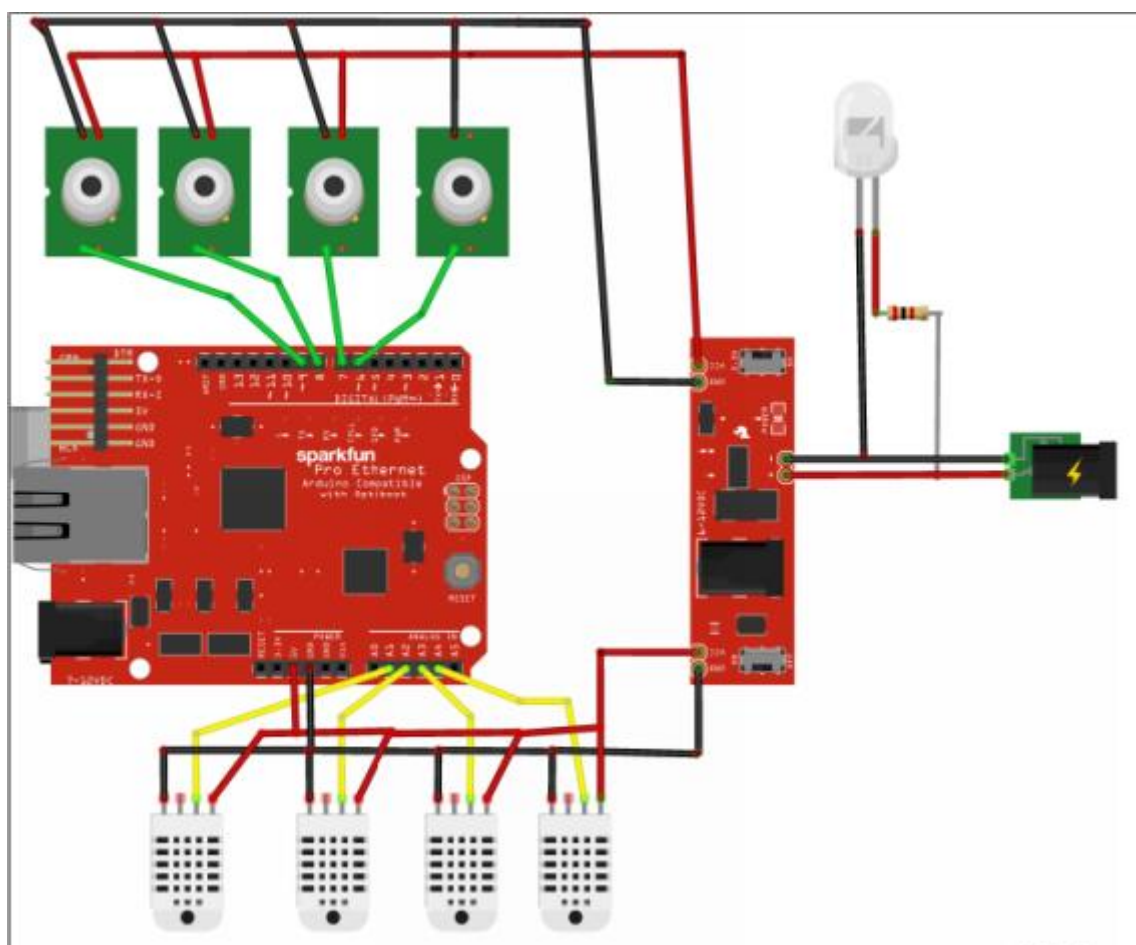
8. Anexos

8.2 Esquemas Eléctricos

8.2.1 Esquema Eléctrico do *Ethernet Shield*



8.2.2 Esquema de ligação dos sensores



8.2.3 Códigos de programação

8.2.3.1 Criação da GUI

8.2.3.1.1 Index.js:

```
$(function(){
    window.setInterval(function(){Grafico();},1000);
    window.setInterval(function(){pegaDados();},1000);
    app.sliders();

});
function PageRequest(param){
    var retorno = "";
    $.ajax({
        type: "GET",
        url: param.url+"?" +param.parm,
        async: false,
        data: { teste: "CONF"}
    })
    .done(function( response ) {
        retorno = response;
    });
    return retorno;
}
function pegaDados(){

    var dados = PageRequest({url:"http://192.168.1.103",parm:""});
    console.log(dados);
    dados = dados.split("|");
    console.log("tempAL:" + dados[7]);
    if(dados[0].trim() == "1")
    {$("#estado1").html("ON");$(".quadroAmb1").css("display","block");}
    else
    {$("#estado1").html("OFF");$(".quadroAmb1").css("display","none");}
    if(dados[1].trim() == "1")
    {$("#estado2").html("ON");$(".quadroAmb2").css("display","block");}
    else
    {$("#estado2").html("OFF");$(".quadroAmb2").css("display","none");}
    if(dados[2].trim() == "1")
    {$("#estado3").html("ON");$(".quadroAmb3").css("display","block");}
    else
    {$("#estado3").html("OFF");$(".quadroAmb3").css("display","none");}
    if(dados[3].trim() == "1")
    {$("#estado4").html("ON");$(".quadroAmb4").css("display","block");}
    else
    {$("#estado4").html("OFF");$(".quadroAmb4").css("display","none");}
```



```

$("#temp1").html(dados[4]+" C");
$("#temp2").html(dados[5]+" C");
$("#temp3").html(dados[6]+" C");
$("#temp4").html(dados[7]+" C");
$("#umid1").html(dados[8]+"%");
$("#umid2").html(dados[9]+"%");
$("#umid3").html(dados[10]+"%");
$("#umid4").html(dados[11]+"%");
var temp = $("#setTemp .tooltip-inner").html();
$("#showSetTemp").html(temp+" °C");

var dados2 =
PageRequest({url:"delayLuz.php",parm:"modo=salvarHorario&amb1="+dados[0].trim()+"&amb2="+d
ados[1].trim()+"&amb3="+dados[2].trim()+"&amb4="+dados[3].trim()});
dados2 = dados2.split("");
var ligarAr1 = 0;
var ligarAr2 = 0;
var ligarAr3 = 0;
var ligarAr4 = 0;
var temPessoa1 = dados2[0];
var temPessoa2 = dados2[1];
var temPessoa3 = dados2[2];
var temPessoa4 = dados2[3];
if(parseInt(temp) < parseInt(dados[4]) && temPessoa1 == "1")
{ligarAr1 = 1;}
if(parseInt(temp) < parseInt(dados[5]) && temPessoa2 == "1")
{ligarAr2 = 1;}
if(parseInt(temp) < parseInt(dados[6]) && temPessoa3 == "1")
{ligarAr3 = 1;}
if(parseInt(temp) < parseInt(dados[7]) && temPessoa4 == "1")
{ligarAr4 = 1;}
console.log("Ar 1: "+ligarAr1);
console.log("Ar 2: "+ligarAr2);
console.log("Ar 3: "+ligarAr3);
console.log("Ar 4: "+ligarAr4);
var dados =
PageRequest({url:"http://192.168.1.103/ligar"+ligarAr1+ligarAr2+ligarAr3+ligarAr4,parm:""}
);
}
function salvarTemp(){
}
function setarTemp(){
}
function Grafico(){
var rotulos = [];
var temp1 = [];
var temp2 = [];
var temp3 = [];
var temp4 = [];
$("#myChart").attr("width","600px");
$("#myChart").attr("height","230px");
var dados1 = PageRequest({url:"dadosTemperatura.php",parm:"modo=amb1"});
dados1 = dados1.split("");

```

```

for(var i = 0; i < dados1.length; i++)
{
var tmp = dados1[i].split(";");
temp1.push(tmp[1]);
rotulos.push(tmp[0]);
}
-2-
C:\wamp64\www\DirectAir\HTML\index.js domingo, 24 de abril de 2016 20:10
var dados2 = PageRequest({url:"dadosTemperatura.php",parm:"modo=amb2"});
dados2 = dados2.split("");
for(var i = 0; i < dados2.length; i++)
{
var tmp = dados2[i].split(";");
temp2.push(tmp[1]);
}
var dados3 = PageRequest({url:"dadosTemperatura.php",parm:"modo=amb3"});
dados3 = dados3.split("");
for(var i = 0; i < dados3.length; i++)
{
var tmp = dados3[i].split(";");
temp3.push(tmp[1]);
}
var dados4 = PageRequest({url:"dadosTemperatura.php",parm:"modo=amb4"});
dados4 = dados4.split("");
for(var i = 0; i < dados4.length; i++)
{
var tmp = dados4[i].split(";");
temp4.push(tmp[1]);
}
var ctx = $("#myChart").get(0).getContext("2d");
var data = {
labels:
[rotulos[0],rotulos[1],rotulos[2],rotulos[3],rotulos[4],rotulos[5],rotulos[6],rotulos[7],r
otulos[8],rotulos[9]],
datasets: [
{
label: "Ambiente 1",
fillColor: "rgba(220,220,220,0.2)",
strokeColor: "rgba(220,220,220,1)",
pointColor: "rgba(220,220,220,1)",
pointStrokeColor: "#fff",
pointHighlightFill: "#fff",
pointHighlightStroke: "rgba(220,220,220,1)",
data:
[temp1[0],temp1[1],temp1[2],temp1[3],temp1[4],temp1[5],temp1[6],temp1[7],temp1[8],
temp1[9]]
},
{
label: "Ambiente 2",
fillColor: "rgba(151,187,205,0.2)",
strokeColor: "rgba(151,187,205,1)",
pointColor: "rgba(151,187,205,1)",
pointStrokeColor: "#fff",
pointHighlightFill: "#fff",

```

```

pointHighlightStroke: "rgba(151,187,205,1)",
data:
[temp2[0],temp2[1],temp2[2],temp2[3],temp2[4],temp2[5],temp2[6],temp2[7],temp2[8],
temp2[9]]
//data: [28, 48, 40, 19, 86, 27, 90]
},
{
label: "Ambiente 3",
fillColor: "rgba(255,0,0,0.2)",
strokeColor: "rgba(255,0,0,1)",
pointColor: "rgba(255,0,0,1)",
pointStrokeColor: "#fff",
pointHighlightFill: "#fff",
pointHighlightStroke: "rgba(151,187,205,1)",
data:
[temp3[0],temp3[1],temp3[2],temp3[3],temp3[4],temp3[5],temp3[6],temp3[7],temp3[8],
temp3[9]]
//data: [28, 48, 40, 19, 86, 27, 90]
},{
label: "Ambiente 4",
fillColor: "rgba(0,255,0,0.2)",
strokeColor: "rgba(0,255,0,1)",
pointColor: "rgba(0,255,0,1)",
pointStrokeColor: "#fff",
pointHighlightFill: "#fff",
pointHighlightStroke: "rgba(151,187,205,1)",
data:
[temp4[0],temp4[1],temp4[2],temp4[3],temp4[4],temp4[5],temp4[6],temp4[7],temp4[8],
temp4[9]]
//data: [28, 48, 40, 19, 86, 27, 90]
}
]
};
var options = {
//Boolean - Whether grid lines are shown across the chart
scaleShowGridLines : true,
//String - Colour of the grid lines
scaleGridLineColor : "rgba(0,0,0,.05)",
//Number - Width of the grid lines
scaleGridLineWidth : 1,
//Boolean - Whether to show horizontal lines (except X axis)
scaleShowHorizontalLines: true,
//Boolean - Whether to show vertical lines (except Y axis)
scaleShowVerticalLines: true,
//Boolean - Whether the line is curved between points
bezierCurve : true,
//Number - Tension of the bezier curve between points
bezierCurveTension : 0.4,
//Boolean - Whether to show a dot for each point
pointDot : true,
//Number - Radius of each point dot in pixels
pointDotRadius : 4,

```

```

//Number - Pixel width of point dot stroke
pointDotStrokeWidth : 1,
//Number - amount extra to add to the radius to cater for hit detection outside the
drawn point
pointHitDetectionRadius : 20,
//Boolean - Whether to show a stroke for datasets
datasetStroke : true,
//Number - Pixel width of dataset stroke
datasetStrokeWidth : 2,
//Boolean - Whether to fill the dataset with a colour
datasetFill : true,
animation: false,
//String - A legend template
legendTemplate : "<ul class=\"<%=name.toLowerCase()>-legend\"><% for (var i=0;
i<datasets.length; i++) { %><li><span
style=\"background-color:<%=datasets[i].strokeColor%>\"></span><%if(datasets[i].label){ %><
%=datasets[i].label%><% } %></li><% } %></ul>"
};
var myLineChart = new Chart(ctx).Line(data, options);
}

```

8.2.3.1.2 DelayLuz.php:

```

<?php
error_reporting(E_WARNING);
$modo = $_GET["modo"];
if($modo == "salvarHorario")
{
$amb1 = $_GET["amb1"];
$amb2 = $_GET["amb2"];
$amb3 = $_GET["amb3"];
$amb4 = $_GET["amb4"];
$servidor = "localhost";
$login = "root";
$senha = "";
$banco = "controlear";
$conexao = mysql_connect($servidor, $login, $senha);
$db = mysql_select_db($banco);
$sql = "select * from ambiente";
$res = mysql_query($sql);
while ($linha = mysql_fetch_array($res)) {
//echo $linha["AmbienteNome"];
}
if(trim($amb1) == "1")
{
$sql = "update ambiente set HoraApagou = null where AmbienteId = 1";
mysql_query($sql);
}
else
{
$sql = "update ambiente set HoraApagou = now() where AmbienteId = 1 and HoraApagou

```

```

is null";
mysql_query($sql);
}
if(trim($amb2) == "1")
{
$sql = "update ambiente set HoraApagou = null where AmbienteId = 2";
mysql_query($sql);
}
else
{
$sql = "update ambiente set HoraApagou = now() where AmbienteId = 2 and HoraApagou
is null";
mysql_query($sql);
}
if(trim($amb3) == "1")
{
$sql = "update ambiente set HoraApagou = null where AmbienteId = 3";
mysql_query($sql);
}
else
{
$sql = "update ambiente set HoraApagou = now() where AmbienteId = 3 and HoraApagou
is null";
mysql_query($sql);
}
-1-

if(trim($amb1) == "4")
{
$sql = "update ambiente set HoraApagou = null where AmbienteId = 4";
mysql_query($sql);
}
else
{
$sql = "update ambiente set HoraApagou = now() where AmbienteId = 4 and HoraApagou
is null";
mysql_query($sql);
}
$ret = "";
$sql = "SELECT TIMESTAMPDIFFF(SECOND,`HoraApagou`,now()) as
Segundos,`AmbienteId`,`AmbienteNome`,`HoraApagou` FROM `ambiente` order by AmbienteId";
$res = mysql_query($sql);
while ($linha = mysql_fetch_array($res)) {
if(trim($ret) != "")
{
$ret .= "|";
}
if($linha["Segundos"] < 60)
{
$ret .= "1";
}
else
{
$ret .= "0";
}
}

```

```

}
//echo $linha["AmbienteNome"];
}
echo $ret;
}
?>

```

8.2.3.1.3 DadosTemperatura.php:

```

<?php
error_reporting(E_WARNING);
$modo = $_GET["modo"];
$servidor = "localhost";
$login = "root";
$senha = "";
$banco = "controlear";
$conexao = mysql_connect($servidor, $login, $senha);
$db = mysql_select_db($banco);
if($modo == "amb1")
{
    $ret = "";
    $sql = "SELECT DATE_FORMAT(TempData,'%H:%i') as Hora, TempGraus FROM Temp where
AmbienteId = 1 order by TempData desc limit 0,10";
    $res = mysql_query($sql);
    while ($linha = mysql_fetch_array($res)) {
        if($ret != ""){$ret .= "|";}
        $ret .= trim($linha["Hora"]).";".trim($linha["TempGraus"]);
    }
    echo $ret;
}
if($modo == "amb2")
{
    $ret = "";
    $sql = "SELECT DATE_FORMAT(TempData,'%H:%i') as Hora, TempGraus FROM Temp where
AmbienteId = 2 order by TempData desc limit 0,10";
    $res = mysql_query($sql);
    while ($linha = mysql_fetch_array($res)) {
        if($ret != ""){$ret .= "|";}
        $ret .= trim($linha["Hora"]).";".trim($linha["TempGraus"]);
    }
    echo $ret;
}
if($modo == "amb3")
{
    $ret = "";
    $sql = "SELECT DATE_FORMAT(TempData,'%H:%i') as Hora, TempGraus FROM Temp where
AmbienteId = 3 order by TempData desc limit 0,10";
    $res = mysql_query($sql);
    while ($linha = mysql_fetch_array($res)) {

```

```

if($ret != ""){ $ret .= "|";}
$ret .= trim($linha["Hora"]).";".trim($linha["TempGraus"]);
}
echo $ret;
}
if($modo == "amb4")
{
$ret = "";
$sql = "SELECT DATE_FORMAT(TempData,'%H:%i') as Hora, TempGraus FROM Temp where
AmbienteId = 4 order by TempData desc limit 0,10";
-1-

$res = mysql_query($sql);
while ($linha = mysql_fetch_array($res)) {
if($ret != ""){ $ret .= "|";}
$ret .= trim($linha["Hora"]).";".trim($linha["TempGraus"]);
}
echo $ret;
}
?>

```

8.2.3.1.4 Salvar.php:

```

<?php
error_reporting(E_WARNING);
// $modo = $_GET["modo"];
$servidor = "localhost";
$login = "root";
$senha = "";
$banco = "controleair";
$conexao = mysql_connect($servidor, $login, $senha);
$db = mysql_select_db($banco);
$sql = "select * from ambiente";
$res = mysql_query($sql);
while ($linha = mysql_fetch_array($res)) {
//echo $linha["AmbienteNome"];
}
// $arquivo =
file_get_contents("http://localhost/DirectAir/HTML/dados.php?modo=pegarDados");
$arquivo = file_get_contents("http://192.168.1.103");
// $arquivo = "1|1|0|1|22|27|26|23|10|20|30|40";
$arquivo = trim($arquivo);
$dados = explode("|", $arquivo);
$sql = "insert into temp(AmbienteId, TempGraus, TempData, TempUmidade)";
$sql .= "values(1, ".$dados[4].", now(), ".$dados[8].")";
mysql_query($sql);
$sql = "insert into temp(AmbienteId, TempGraus, TempData, TempUmidade)";
$sql .= "values(2, ".$dados[5].", now(), ".$dados[9].")";
mysql_query($sql);

```

```
$sql = "insert into temp(AmbienteId, TempGraus, TempData,TempUmidade)";  
$sql .= "values(3, ".$dados[6].",now(), ".$dados[10].")";  
mysql_query($sql);  
$sql = "insert into temp(AmbienteId, TempGraus, TempData,TempUmidade)";  
$sql .= "values(4, ".$dados[7].",now(), ".$dados[11].")";  
mysql_query($sql);  
?>
```


8.2.3.2 Arduino Mega

```
#include <dht11.h>

#include "DHT.h"
#include <SPI.h>
#include <Ethernet.h> // Biblioteca utilizada para comunicação com o Arduino

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
byte ip[] = { 192, 168, 1, 103 }; //Define o endereço IP.
EthernetServer server(80); // Porta onde estará aberta para comunicação Internet e Arduino.
String readString;
String valPag;

#define DHTPIN1 A1 // pino que conecta o sensor de temperatura e umidade 1
#define DHTPIN2 A2 // pino que conecta o sensor de temperatura e umidade 2
#define DHTPIN3 A3 // pino que conecta o sensor de temperatura e umidade 3
#define DHTPIN4 A4 // pino que conecta o sensor de temperatura e umidade 4

#define DHTTYPE DHT11 // DHT 11

DHT dht(DHTPIN1, DHTTYPE);
DHT dht2(DHTPIN2, DHTTYPE);
DHT dht3(DHTPIN3, DHTTYPE);
DHT dht4(DHTPIN4, DHTTYPE);

int pinopir1 = 7; //Pino ligado ao sensor PIR
int pinopir2 = 8; //Pino ligado ao sensor PIR2
int pinopir3 = 9; //Pino ligado ao sensor PIR3
int pinopir4 = 6; //Pino ligado ao sensor PIR4

int presenca1 = 0;
int presenca2 = 0;
int presenca3 = 0;
int presenca4 = 0;

int temp1 = 0;
int temp2 = 0;
int temp3 = 0;
int temp4 = 0;

int umid1 = 0;
int umid2 = 0;
int umid3 = 0;
int umid4 = 0;

void setup()
{
```

```
Ethernet.begin(mac, ip); // Chama o MAC e o endereço IP da placa Ethernet.  
server.begin(); // Inicia o servidor que esta inserido junto a placa Ethernet.
```

```
pinMode(pinorele1, OUTPUT); //Define pino rele como saida  
pinMode(pinopir1, INPUT); //Define pino sensor como entrad  
pinMode(pinopir2, INPUT); //Define pino sensor como entrad  
pinMode(pinopir3, INPUT); //Define pino sensor como entrad
```

```
pinMode(2, OUTPUT);  
pinMode(3, OUTPUT);  
pinMode(4, OUTPUT);  
pinMode(5, OUTPUT);
```

```
Serial.begin(9600);  
// Serial.println("Verificacao de Temperatura ");
```

```
dht.begin();
```

```
}
```

```
void loop()
```

```
{
```

```
  EthernetClient client = server.available();
```

```
  if (client) {
```

```
    while (client.connected()) {
```

```
      if (client.available()) {
```

```
        char c = client.read();
```

```
        valPag.concat(c);
```

```
        if (readString.length() < 100) {
```

```
          readString += c;
```

```
        }
```

```
        if (c == '\n') {
```

```
          client.println("HTTP/1.1 200 OK");
```

```
          client.println("Content-Type: text/html");
```

```
          client.println("Access-Control-Allow-Origin: *");
```

```
          client.println();
```

```
          float t = dht.readTemperature();
```

```
          float t2 = dht2.readTemperature();
```

```
          float t3 = dht3.readTemperature();
```

```
          float t4 = dht4.readTemperature();
```

```
          float h1 = dht.readHumidity();
```

```
          float h2 = dht2.readHumidity();
```

```
          float h3 = dht3.readHumidity();
```

```

float h4 = dht4.readHumidity();

temp1 = t;
temp2 = t2;
temp3 = t3;
temp4 = t4;

umid1 = h1;
umid2 = h2;
umid3 = h3;
umid4 = h4;

presenca1 = 0;
presenca2 = 0;
presenca3 = 0;
presenca4 = 0;

presenca1 = digitalRead(pinopir1);
presenca2 = digitalRead(pinopir2);
presenca3 = digitalRead(pinopir3);
presenca4 = digitalRead(pinopir4);

client.println(presenca1);
client.println("|");
client.println(presenca2);
client.println("|");
client.println(presenca3);
client.println("|");
client.println(presenca4);
client.println("|");
client.println(temp1);
client.println("|");
client.println(temp2);
client.println("|");
client.println(temp3);
client.println("|");
client.println(temp4);
client.println("|");
client.println(umid1);
client.println("|");
client.println(umid2);
client.println("|");
client.println(umid3);
client.println("|");
client.println(umid4);

delay(1);
client.stop();

if(valPag.indexOf("ligar") > 0)
{
    valPag = valPag.substring(valPag.indexOf("ligar") + 5, valPag.indexOf("ligar") + 9);
}

```

```
Serial.println(valPag);

if(valPag.substring(0,1) == "1")
{
  digitalWrite(2,HIGH);
}
else
{
  digitalWrite(2,LOW);
}

if(valPag.substring(1,2) == "1")
{
  digitalWrite(3,HIGH);
}
else
{
  digitalWrite(3,LOW);
}

if(valPag.substring(2,3) == "1")
{
  digitalWrite(4,HIGH);
}
else
{
  digitalWrite(4,LOW);
}

if(valPag.substring(3,4) == "1")
{
  digitalWrite(5,HIGH);
}
else
{
  digitalWrite(5,LOW);
}

}

valPag = "";

}
}
}
}
}
```