



TRABALHO DE GRADUAÇÃO

**AUTOMAÇÃO COM OPENHAB DE SISTEMA DE
CLIMATIZAÇÃO HÍBRIDA VISANDO
EFICIÊNCIA ENERGÉTICA**

Daniel Bello Boaventura

Brasília, julho de 2016

UNIVERSIDADE DE BRASÍLIA

**FACULDADE DE TECNOLOGIA
CURSO DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE E
AUTOMAÇÃO**

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia
Curso de Graduação em Engenharia de Controle e Automação

TRABALHO DE GRADUAÇÃO

AUTOMAÇÃO COM OPENHAB DE SISTEMA DE CLIMATIZAÇÃO HÍBRIDA VISANDO EFICIÊNCIA ENERGÉTICA

Daniel Bello Boaventura

Relatório submetido como requisito parcial para obtenção
do grau de Engenheiro de Controle e Automação.

Banca Examinadora

Prof. Adolfo Bauchpiess, UnB/ ENE (Orientador) _____

Prof. Eugênio L. F. Fortaleza, UnB/ ENM _____

Prof. João Manoel Dias Pimenta, UnB/ ENM _____

Brasília, julho de 2016

FICHA CATALOGRÁFICA

BOAVENTURA, DANIEL BELLO	
Automação com openHAB de Sistema de Climatização Híbrida visando Eficiência Energética, [Distrito Federal] 2016.	
xvii, 123p., 297 mm (FT/UnB, Engenheiro, Controle e Automação, Ano). Trabalho de Graduação – Universidade de Brasília.Faculdade de Tecnologia.	
1.Sistema Híbrido de Climatização	2.Resfriamento Evaporativo
3.OpenHAB	
I. Mecatrônica/FT/UnB	II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

BOAVENTURA, D.B., (ANO). Automação com openHAB de Sistema de Climatização Híbrida visando Eficiência Energética. Trabalho de Graduação em Engenharia de Controle e Automação, Publicação FT.TG-nº , Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF, 72p.

CESSÃO DE DIREITOS

AUTOR: Daniel Bello Boaventura.

TÍTULO DO TRABALHO DE GRADUAÇÃO: Automação com openHAB de Sistema de Climatização Híbrida visando Eficiência Energética.

GRAU: Engenheiro

ANO: 2016

É concedida à Universidade de Brasília permissão para reproduzir cópias deste Trabalho de Graduação e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desse Trabalho de Graduação pode ser reproduzida sem autorização por escrito do autor.

Daniel Bello Boaventura
Quadra 204, Lote 8, Bloco A ap 602 – Águas Claras.
71939-540 Brasília – DF – Brasil.

AGRADECIMENTOS

À Deus por me dar forças e sabedoria.

*Aos meus pais Jairo e Isabel por todos os ensinamentos e construírem as bases para que
hoje pudesse chegar até aqui.*

Ao meu irmão Lucas pela paciência e ajuda.

À minha linda Amanda por todo o incentivo e confiança.

RESUMO

O presente texto trata do estudo e projeto da lógica de controle para sistemas de climatização híbrida, que consistem na junção de um sistema de refrigeração convencional com um sistema de climatização evaporativo. O projeto tem por objetivo a priorização do sistema de climatização evaporativa, uma vez que esse sistema consome menos energia para promover o controle da temperatura. O sistema de automação foi desenvolvido com o software livre openHAB e o protocolo de comunicação entre as partes do sistema foi o MQTT, também software livre.

Palavras Chave: Ar Condicionado; Racionalização de Energia; Sistema de Automação; Sistema de Controle Supervisório; Controle de Processos; openHAB; MQTT; Resfriamento Evaporativo; Sistema Híbrido;

ABSTRACT

This paper deals with the study and control logic design for hybrid air conditioning systems, which consist of the addition of a conventional cooling system with an evaporative cooling system. The project aims to prioritize the evaporative cooling system, since this system uses less energy to promote temperature control. The automation system was developed with open source software openHAB and the communication protocol between the parts of the system was the MQTT also open source software.

Keywords: Air Conditioning; Energy Rationalization; Automation System; Supervisory Control System; Process Control; openHAB; MQTT; Evaporative Cooler; Hybrid System;

SUMÁRIO

CAPÍTULO 1 - INTRODUÇÃO	12
1.1 MOTIVAÇÃO.....	12
1.2 OBJETIVOS.....	12
CAPÍTULO 2 – Fundamentos Teóricos.....	13
2.1. PSICROMETRIA.....	13
2.1.1 AR ATMOSFÉRICO, AR SECO E AR ÚMIDO	13
2.2 PROPRIEDADES PSICOMÉTRICAS.....	14
2.2.1 TEMPERATURA DE BULBO SECO.....	14
2.2.2 TEMPERATURA DE BULBO ÚMIDO	15
2.2.3 UMIDADE ABSOLUTA.....	15
2.2.4 UMIDADE RELATIVA	15
2.2.5 TEMPERATURA DE PONTO DE ORVALHO.....	16
2.3 CARTA PSICROMÉTRICA.....	16
2.4 RESFRIAMENTO EVAPORATIVO DO AR	18
2.5 SISTEMA CONVENCIONAL DE REFRIGERAÇÃO.....	20
2.6 SISTEMA DE CLIMATIZAÇÃO HÍBRIDA	21
2.7 SISTEMA SUPERVISÓRIO OPENHAB	21
2.7.1 ARQUITETURA.....	21
2.7.1.1 OPENHAB RUNTIME	21
2.7.1.2 OPENHAB DESIGNER.....	23
2.7.2 LIGAÇÕES (BIDINGS).....	24
2.7.3 BANCO DE DADOS.....	25
2.7.4 REGRAS	25
2.8 PROTOCOLO DE COMUNICAÇÃO MQTT.....	26
2.8.1 ECLIPSE MOSQUITTO BROKER	27
2.9 CARGA ELÉTRICA E CONSUMO DE ENERGIA ELÉTRICA.....	28
CAPÍTULO 3 – SISTEMA HÍBRIDO DE CLIMATIZAÇÃO DA SALA DE REUNIÕES DO LARA.....	29
3.1 ASPECTOS CONSTRUTIVOS.....	29
3.2 QUADRO DE COMANDO.....	32
3.3 SENSORIAMENTO.....	34
3.3.1 SENSOR DE TEMPERATURA E UMIDADE DHT22	34
3.3.2 CALIBRAÇÃO DO SENSOR DHT22	35
3.4 VISÃO GERAL DO SISTEMA DE AUTOMAÇÃO.....	36
3.4.1 RASBERRY PI	37
3.4.2 ARDUINO.....	38
3.4.3 MÓDULO ETHERNET ENC 28J60.....	39
3.4.4 MÓDULO RELÉ	40
3.4.5 Roteador	42
3.5 IMPLEMENTAÇÃO DO PROTOCOLO DE COMUNICAÇÃO	42
3.6 PARÂMETROS DE PROJETO.....	45
3.6.1 CÁLCULO DE CONSUMO POR UNIDADE	45
3.7 DESCRIÇÃO DE FUNCIONAMENTO	45
3.7.1 IMPLEMENTAÇÃO DA LÓGICA DE AUTOMAÇÃO COM BASE NA TEMPERATURA DE BULBO ÚMIDO EXTERNA.....	46
CAPÍTULO 4 – RESULTADOS E ANÁLISE.....	47
4.1 TEMPERATURA DE BULBO ÚMIDO EXTERNA.....	47

4.1. LÓGICA DE AUTOMAÇÃO IMPLEMENTADA POR OLMOS	50
4.2 LÓGICA DE AUTOMAÇÃO COM BASE NA TEMPERATURA DE BULBO ÚMIDO DO AR EXTERNO	53
CAPÍTULO 5 – CONCLUSÕES	56
5.1. CONCLUSÕES GERAIS.....	56
5.1. PERSPECTIVAS PARA PESQUISAS FUTURAS	57
Anexo 1 – Códigos de Configuração do openHAB	60
Anexo 2 – Código de Configuração do Arduino	66

LISTA DE FIGURAS

Figura 1. Carta psicrométrica de Brasília	17
Figura 2. Coeficientes de correlação.....	18
Figura 3. (a) Resfriamento evaporativo Direto. (b) Resfriamento evaporativo Indireto	19
Figura 4. Arquitetura do openHAB	22
Figura 5. Barramento de eventos do openHAB.....	23
Figura 6. openHAB Designer.....	24
Figura 7. Arquitetura do paradigma publish/subscribe	27
Figura 8. Projeto do Sistema Híbrido de Climatização da Sala de Reuniões do LARA	29
Figura 9. Sistema de refrigeração <i>split</i> Versatile dutável da Carrier – 18.000 BTUs	30
Figura 10. Dispensador de água de celulose da Munters usado no sistema evaporativo	31
Figura 11. Bomba d'Água Átoma 8W para dispersão de água na manta de celulose.	31
Figura 12. Atuador para <i>dampers</i> , modelo AM-2001, 24 Vac x 0 ~ 10Vcc.....	32
Figura 13. Placa eletrônica DRT-3403-N, 24 Vac x 0 ~ 10Vcc	32
Figura 14. Diagrama de blocos unifilar para o Sistema Híbrido de Climatização da Sala de Reuniões do LARA.....	33
Figura 15. Sensor de temperatura e umidade DHT22	34
Figura 16. Estação de monitoramento de temperatura e umidade 3M Quest Temp° 36©.....	36
Figura 17. Diagrama Geral do Sistema de Automação.....	37
Figura 18. Raspberry Pi 2 Model B	38
Figura 19. Arduino Mega 2560	39
Figura 20. Módulo Ethernet ENC 28J60.....	40
Figura 21. Desenho esquemático de um relé.....	41
Figura 22. Módulo Relé 5V com 4 relés usado no projeto	41
Figura 23. Roteador sem fio de 4 portas ethernet (RJ45)	42
Figura 24. Verificação dos estados serviços openHAB, Mosquitto e MySQL. O símbolo “[+]” indica que o serviço está sendo executado	42
Figura 25. Diagrama de comunicação com o protocolo MQTT	44
Figura 26. Diagrama de Fluxo da Lógica de Automação do Sistema de Ar Condicionado Híbrido da Sala de Reuniões do LARA	46
Figura 27. Comportamento das temperaturas externas de bulbo seco (Temperatura Externa – em vermelho) e de bulbo úmido (TBU Externa – em Amarelo) durante 5 dias.....	47
Figura 28. Comportamento das temperaturas externas de bulbo seco (Temperatura Externa – em vermelho) e de bulbo úmido (TBU Externa – em Amarelo) durante um dia. A TBU Externa ultrapassa a barreira dos 18°C às 7h10. Essa temperatura volta a ficar abaixo dos 18°C em torno de 14h.	48
Figura 29. Comportamento das temperaturas externas de bulbo seco (Temperatura Externa – em vermelho) e de bulbo úmido (TBU Externa – em Amarelo) durante um dia. A TBU Externa ultrapassa a barreira dos 18°C em torno das 7h. Essa temperatura volta a ficar abaixo dos 18°C em torno de 14h10.	48
Figura 30. Comportamento da umidade externa e das temperaturas externas de bulbo seco (Temperatura Externa – em vermelho) e de bulbo úmido (TBU Externa – em Amarelo) durante 5 dias.....	49

Figura 31. Comportamento da variação da temperatura de bulbo seco interna (Temperatura Interna – em vermelho) quando se mantém somente a bomba d’água ligada. Nesse intervalo, a maior diferença entre a temperatura interna e a temperatura de bulbo úmido externa foi de 4,5°C.....	49
Figura 32. Lógica de controle implementada por Olmos em sua tese de mestrado.	50
Figura 33. Gráfico da Temperatura e Umidade do ar Externo entre os dias 23/06/2009 e 24/06/2009	51
Figura 34. Gráfico da Temperatura e Umidade do ar Externo entre os dias 05/07/2016 e 06/07/2016.	51
Figura 35. Interface gráfica do openHAB mostrando o sistema de refrigeração convencional ligado, mesmo com a TBU do ar externo em 17,1°C.....	52
Figura 36. Detalhes do medidor de energia <i>Landis + Gyr</i>	52
Figura 37. Implementação da lógica de automação com base na TBU do ar externo	53
Figura 38. Interface gráfica do openHAB mostrando o sistema de refrigeração evaporativo ligado e controlando a temperatura interna	54
Figura 39. Interface para telefone celular do openHAB mostrando situação em que a TBU Externa está maior do que 19,5 e a temperatura interna está maior que 23,6. Nesse caso, o sistema convencional é ligado e o evaporativo, desligado.	55
Figura 40. Situação em que a TBU Externa aumentou e o sistema convencional teve que ser ligado.	55

LISTA DE TABELAS

Tabela 1. Composição do ar seco na troposfera.....	13
Tabela 2. Redução da temperatura interna em função da umidade relativa do ar externo e sua temperatura	20
Tabela 3. Comparação entre MQTT e HTTP	26
Tabela 4. Lista de materiais utilizados na implementação do sistema de ar condicionado híbrido da sala de reuniões do LARA, juntamente com a legenda dos itens apontados na Figura 8	30
Tabela 5. Pinos do sensor DHT22	34
Tabela 6. Especificações técnicas do sensor DHT22	35
Tabela 7. Lista de equipamentos do sistema de automação	36
Tabela 8. Especificações técnicas do Arduino Mega 2560	39
Tabela 9. Consumo por equipamento	45
Tabela 10. Parâmetros de controle do sistema de climatização híbrida	50
Tabela 11. Comparação de Consumo do sistema no ano de 2009 e 2016	53

CAPÍTULO 1 - INTRODUÇÃO

Essa primeira seção abordará uma visão geral do objetivo desse trabalho, seus aspectos gerais e suas implicações. O tema abordado envolve as bases de funcionamento do sistema híbrido instalado na sala de reuniões do LARA, no qual esse trabalho foi baseado. Os objetivos e resultados esperados serão explanados.

1.1 MOTIVAÇÃO

Estudos nacionais e internacionais apontam que a sociedade não poderá prescindir de aumentar sua eficiência energética, como parte atuante da estratégia de atendimento à demanda energética, que sempre tende ao crescimento. A eficiência energética tem papel preponderante na segurança energética de um país, além de contribuir para a modicidade tarifária, competitividade da economia e redução de impactos ambientais.

Considerando-se que 40% do consumo energético brasileiro em 2012 foi gasto em edifícios e que o sistema de ar condicionado é a maior carga em edifícios (pode chegar em 62,9% do consumo total), aumentar a eficiência energética desse sistema terá grande impacto na eficiência energética do país como um todo. Dessa forma, a redução em 50% do consumo energético dos aparelhos de ar condicionado do país fará com que a eficiência energética desse aumente em 20%, o que mostra o grande impacto dessa redução. Com o aumento da eficiência energética do país, pode-se diminuir ou até mesmo retardar o crescimento da demanda energética.

1.2 OBJETIVOS

O objetivo geral desse trabalho é achar um ponto de operação do sistema de ar condicionado híbrido instalado na sala de reuniões do LARA, com a finalidade de reduzir ainda mais o seu consumo de energia elétrica. Esse ponto de operação corresponde na forma de funcionamento ideal dos dois sistemas do ar condicionado (sistema de refrigeração convencional do tipo *split* e sistema de climatização evaporativa) funcionando isoladamente ou em conjunto para manter-se o controle da temperatura desejada e consumir o mínimo possível de energia. Segundo a dissertação de mestrado de Olmos [1], mostrou-se que definindo-se *set-points* para ligar ou desligar os dois sistemas, foi alcançada uma economia de 67,1%. Assim, esse trabalho objetiva aumentar ainda mais a economia de energia.

Tem-se como objetivo específico o estudo aprofundado do sistema de climatização evaporativo de modo que esse sistema seja priorizado pelo sistema de automação do ar condicionado híbrido da sala de reuniões do LARA, ou seja, o controle da temperatura para conforto térmico deve ser alcançado com o mínimo de uso do sistema de refrigeração convencional do tipo *split*.

CAPÍTULO 2 – Fundamentos Teóricos

2.1. PSICROMETRIA

Psicrometria é a parte da termodinâmica que tem como objetivo o estudo das propriedades do ar úmido, o que consiste nas interações entre calor, umidade e ar seco. Dessa forma, a psicrometria é essencial para o projeto e para a análise de sistemas de ar condicionado, desumidificação e resfriamento evaporativo, onde o controle das condições em que esse ar se encontra é essencial para o seu maior rendimento [2].

2.1.1 AR ATMOSFÉRICO, AR SECO E AR ÚMIDO

O ar atmosférico é composto por gases, vapor d'água e contaminantes, como poeira, pólen, resíduos etc. O ar seco é o que resta do ar atmosférico quando todo o vapor d'água e todos os contaminantes são removidos. A composição do ar seco é relativamente constante, porém, pequenas variações nas quantidades de um componente individual podem ocorrer em função da hora do dia, da localização geométrica e da altitude [3]. A Tabela 2.1 apresenta a composição aproximada do ar seco em termos de percentuais volumétricos de seus componentes. A massa molecular do ar seco na escala carbono-12 é de 28,9645. Assim, a constante do gás para o ar seco é $R_a = 8314,41/28,9645 = 287,055 \text{ J}/(\text{KgK})$.

Tabela 1. Composição do ar seco na troposfera [3]

Componente	Volume (%)
Nitrogênio	78,084
Oxigênio	20,9476
Argônio	0,934
Dióxido de Carbono	0,0314
Neônio	0,001818
Hélio	0,000524
Metano	0,00015
Dióxido de Enxofre	0 até 0,001
Hidrogênio	0,00005
Kriptônio, Xenônio e Ozônio	0,0002

O ar úmido é a mistura contendo ar seco e vapor d'água. A quantidade de vapor d'água varia de zero até um máximo, que é função da temperatura e da pressão da mistura. Essa última condição é conhecida como ar úmido saturado. Nessa condição, há equilíbrio térmico entre o vapor d'água e o seu condensado. A massa molecular da água, na escala do carbono-12, é de 18,01528. Assim, a constante do gás para o vapor d'água é $R_v = 8314,41/18,01528 = 461,520 \text{ J}/(\text{KgK})$.

A temperatura e a pressão barométrica do ar atmosférico variam consideravelmente com a altitude, a localização geográfica e a condição climática. O ar atmosférico padrão é uma referência para estimar as propriedades do ar em várias altitudes. Ao nível do mar, a temperatura de 20°C e a pressão barométrica de 101,325 kPa são os valores padrões. Assim, em altitudes entre 500 e 11000 metros, as temperaturas e pressões atmosféricas (barométrica) são dadas por:

$$t = 20 - 0,0065H \quad (1)$$

$$P_{atm} = 101,325(1 - 2,25577 \times 10^{-5}H)^{5,2559} \quad (2)$$

onde,

t = temperatura [°C]

P_{atm} = pressão atmosférica [kPa]

H = altitude [m]

2.2 PROPRIEDADES PSICOMÉTRICAS

Em psicrometria, os componentes do ar que apresentam temperatura de condensação muito baixa são reunidos em uma única fase denominada ar seco, enquanto o vapor d'água, que condensa em condições típicas encontradas em sistemas de condicionamento de ar, é tratado independentemente. Desse modo, admite-se o ar úmido como sendo a mistura de dois gases ideais: ar seco e vapor d'água. As leis das pressões parciais, dos volumes parciais e a equação de estado do gás ideal são usadas para estabelecer as equações que definem os estados psicrométricos do ar. Entretanto as seguintes hipóteses são admitidas:

- A fase líquida (vapor d'água) não contém gases dissolvidos
- A fase gasosa (ar seco) pode ser tratada como uma mistura de gases ideais
- Quando a mistura e o condensado (água líquida) estão em uma dada pressão e temperatura, o equilíbrio entre o condensado e seu vapor não é afetado pela presença de outros componentes, ou seja, quando o equilíbrio é alcançado, a pressão parcial do vapor é igual à pressão de saturação correspondente à temperatura da mistura.

Portanto, admite-se que, na mistura, a temperatura do ar seco é igual à do vapor d'água.

2.2.1 TEMPERATURA DE BULBO SECO

A temperatura de bulbo seco (TBS) refere-se a temperatura do ar ambiente medida por um termômetro convencional exposto livremente ao ar, mas protegido de radiação e umidade. É chamada de "bulbo seco" porque a temperatura indicada pelo termômetro não é influenciada pela umidade do ar.

A temperatura de bulbo seco é um indicativo da quantidade de calor contida no ar. Quando a temperatura de bulbo seco aumenta, a capacidade de absorção de umidade do ar também aumenta. Usualmente, sua unidade de representação é o grau Celsius [°C], porém sua unidade SI é o Kelvin [K].

2.2.2 TEMPERATURA DE BULBO ÚMIDO

A temperatura de bulbo úmido (TBU) é a temperatura medida por um termômetro que tem o seu bulbo coberto por uma malha porosa (normalmente algodão), imersa em um recipiente contendo água destilada. Esse bulbo umedecido deve ficar exposto a uma corrente de ar para favorecer a evaporação da água, o que causa a diminuição da temperatura indicada pelo termômetro, uma vez que a evaporação consome calor. Quanto menor é a presença de umidade no ar, maior será a evaporação da água presente na malha e, conseqüentemente, menor será a temperatura de bulbo úmido. Assim, a temperatura de bulbo úmido será sempre menor ou igual à temperatura de bulbo seco, uma vez que não haverá evaporação quando o ar encontra-se saturado. Pode-se afirmar que a temperatura de bulbo úmido é um indicador da quantidade de umidade contida no ar e representa a temperatura mais baixa que pode ser alcançada evaporando-se a água.

A temperatura de bulbo úmido é bastante útil para o processo de resfriamento evaporativo, uma vez que o seu rendimento é inversamente proporcional à temperatura de bulbo úmido. Assim como a temperatura de bulbo seco, a sua unidade de medida é o grau Celsius [°C].

Segundo [7], a temperatura de bulbo úmido pode ser calculada usando-se a Equação (3).

$$TBU = TBS \times \text{atan} \left[0,151977(\phi_{ar} + 8,313659)^{1/2} \right] + \text{atan}[TBS + \phi_{ar}] - \text{atan}[\phi_{ar} - 1,676331] + 0,00391838(\phi_{ar})^{3/2} \text{atan}[0,023101\phi_{ar}] - 4,686035 \quad (3)$$

A Equação (3) é válida para pressão atmosférica de 1013,25 hPa e temperaturas variando entre -20°C e 50°C.

Como comparativo, para uma temperatura de bulbo seco de 25°C e umidade relativa do ar de 50%, usando-se a carta psicrométrica de Brasília, que foi elaborada para um pressão de 893,63 hPa, a temperatura de bulbo úmido será de 17,7°C. Usando-se a Equação 3, a temperatura de bulbo úmido será que 17,9°C. Como o erro produzido pelo uso da Equação 3 é de 1,13%, pode-se dizer que, para efeitos de conforto térmico, esse erro é irrelevante.

2.2.3 UMIDADE ABSOLUTA

A umidade absoluta, w_{abs} , descreve a quantidade de vapor d'água dissolvido em 1 Kg de ar seco, ou seja, é uma relação entre a massa de vapor d'água, m_w , e a massa de ar seco, m_a . Usualmente é medida em Kg de vapor d'água por Kg de ar seco, [3].

$$w_{abs} = \frac{m_w}{m_a} = \frac{\text{Kg de vapor d'água}}{\text{Kg de ar seco}} \left[\frac{\text{Kg}_v}{\text{Kg}_a} \right] \quad (4)$$

2.2.4 UMIDADE RELATIVA

A umidade relativa, ϕ_{ar} , é a porcentagem que representa a razão entre a massa atual de umidade no ar, m_{vapH_2O} comparada com o total ou o máximo de umidade que o ar consegue suportar para dada temperatura, m_{mvap} [3].

$$\phi_{ar} = \frac{m_{vapH_2O}}{m_{mvap}} \times 100 \quad (5)$$

2.2.5 TEMPERATURA DE PONTO DE ORVALHO

A temperatura de ponto de orvalho, T_d , é a temperatura na qual o vapor d'água começa a condensar e o ar torna-se completamente saturado. Acima dessa temperatura, a umidade continuará no ar. Abaixo dela, o vapor d'água condensará.

A temperatura de ponto de orvalho, T_d , é expressa em graus Celsius (°C) e também pode representar uma medida absoluta da umidade no ar à uma determinada pressão. Assim, se a temperatura de ponto de orvalho é próxima da temperatura do ar, a umidade relativa do ar está alta.

Segundo [8], a temperatura de ponto de orvalho pode ser calculada usando-se a temperatura de bulbo seco (TBU) e a umidade relativa do ar, conforme a Equação 8, conhecida como Equação de Arden Buck.

$$E_s = 6,112 \times e^{(17,67TBS)/(TBS+243,5)} \quad (6)$$

$$E = \frac{E_s \times \phi_{ar}}{100} \quad (7)$$

$$T_d = \frac{[243,5 \times \ln(E/6,112)]}{17,67 - \ln(E/6,112)} \quad (8)$$

Onde,

E_s = Pressão de Saturação do Vapor d'Água

TBS = Temperatura de Bulbo Seco

ϕ_{ar} = Umidade relativa do ar

E = Pressão do Vapor d'Água

T_d = Temperatura de Ponto de Orvalho

A precisão mínima obtida com o uso da Equação 8 é de 0,1% para

$$-30^\circ\text{C} \leq TBS \leq 30^\circ\text{C}$$

$$1\% < \phi_{ar} < 100\%$$

Pode afirmar que:

- Quando o ar esquentar, sua umidade relativa diminuirá
- Quando o ar esfriar, sua umidade relativa aumentará
- Quando o ar esfriar em temperaturas iguais ou menores do que a temperatura de ponto de orvalho, a água condensa e seu excedente é eliminado, tornando o ar mais seco.

2.3 CARTA PSICROMÉTRICA

As interações entre a umidade do ar e o calor são bastante complexas. Para melhor representar essas interações, usa-se um diagrama denominado Carta Psicrométrica. Esse gráfico permite obter o traçado de diversos processos psicrométricos e a determinação de estados e propriedades do ar úmido. É construído com base no fato de que o estado termodinâmico do ar úmido é determinado por três propriedades independentes. Assim, se uma das três for mantida constante, as outras duas formam os eixos da carta psicrométrica. Qualquer ponto do gráfico definirá um estado da mistura, ou seja, do ar úmido. Geralmente a pressão da

mistura é escolhida como propriedade a ser mantida constante, já que na maioria dos processos psicrométricos sua variação é desprezível [9].

De fato, todas as cartas psicrométricas são construídas para uma pressão de mistura fixa. Do ponto de vista estritamente termodinâmico, quaisquer outras duas propriedades poderiam ser usadas na construção dos eixos. O diagrama que adota a entalpia específica (h) e a umidade absoluta (W) como eixos e com pressão constante é chamado diagrama de Mollier.

Como foi dito, as cartas psicrométricas são elaboradas para uma pressão atmosférica constante. Por isso, cada região ou cidade tem o seu próprio diagrama. A Figura 1 mostra a carta psicrométrica da cidade de Brasília.

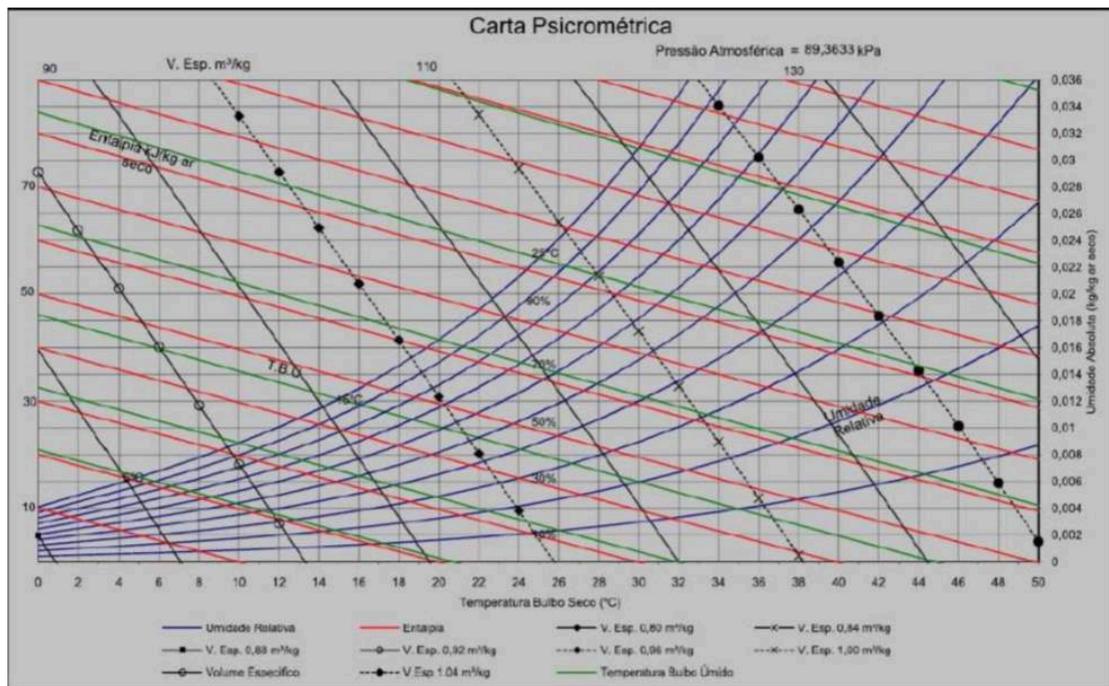


Figura 1. Carta psicrométrica de Brasília

As propriedades do ar relacionadas na carta psicrométrica são:

- Temperatura de bulbo seco
- Temperatura de bulbo úmido
- Umidade absoluta
- Umidade relativa
- Volume específico
- Entalpia específica

As propriedades citadas podem ser vistas de forma esquemática na Figura 2, que representa uma carta psicrométrica típica. Nessa figura, os seguintes pontos são mostrados:

1. Temperatura de Bulbo Seco, TBS, indicada por linhas verticais e expressa em graus Celsius (°C)
2. Umidade Absoluta, w_{abs} , representada por linhas horizontais e expressa em Kg_v/Kg_a
3. Escala de umidade absoluta

4. Temperatura de Bulbo Úmido, TBU, indicada por linhas transversais e escala na extremidade esquerda da carta. É expressa em graus Celsius (°C)
5. Volume específico, V, expresso em m^3/Kg_a
6. Escala da entalpia específica, expressa em kJ/Kg_a de ar seco
7. Temperatura de ponto de orvalho, T_d , na linha de saturação e em graus Celsius (°C)
8. Umidade Relativa, ϕ_{ar} , expressa em porcentagem
9. Escala referente ao fator de calor sensível (FCS).

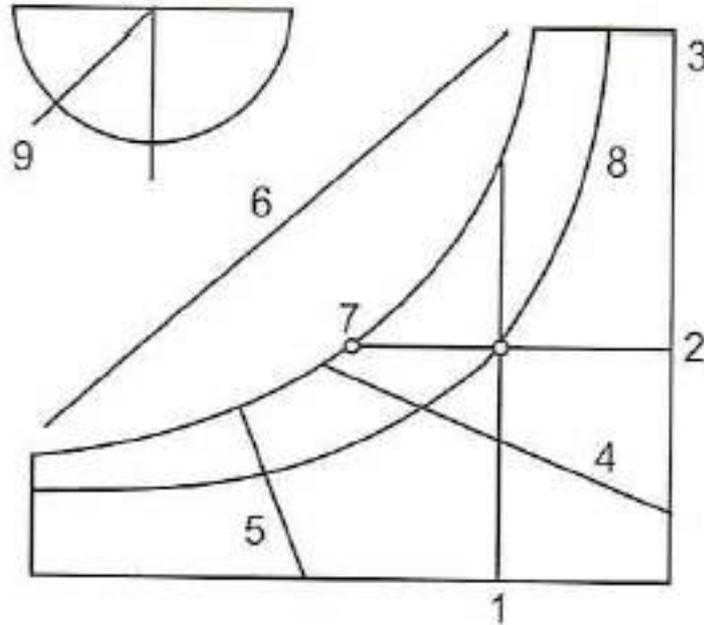


Figura 2. Coeficientes de correlação.

2.4 RESFRIAMENTO EVAPORATIVO DO AR

Refrigeradores evaporativos de ar são sistemas que usam apenas água e um ventilador para circular o ar. Quando o ar quente e seco (não saturado) é forçado a passar através de uma manta encharcada, a água é evaporada e esse vapor é absorvido pelo ar. Nesse processo de evaporação induzida, o ar cede energia (calor) para que a água evapore, o que resulta em um ar mais frio e úmido na saída do evaporador.

Sistemas de resfriamento evaporativos são uma alternativa mais eficiente em termos de energia se comparados aos sistemas de resfriamento baseados em compressores. O seu funcionamento é muito mais efetivo em cidades de clima seco e requer uma fonte abundante de água.

Sobre a efetividade do sistema de resfriamento evaporativo, sua eficiência pode ser calculada pela Equação 9 [10].

$$\eta_{re} = \frac{T_{ENTRADA} - T_{SAÍDA}}{T_{ENTRADA} - T_{BU\text{ EXTERNA}}} \quad (9)$$

Da Equação 9 observa-se que quanto menor a TBU do ar externo, maior será a eficiência do sistema.

Segundo as leis de cinética química, dois fatores, entre outros, influenciam na velocidade de uma reação:

- A concentração dos reagentes

- A superfície de contato entre os reagentes

Assim, quanto menor é quantidade de umidade no ar (concentração), ao forçar-se a passagem do ar por uma superfície molhada, maior será a velocidade de transferência de umidade para o ar. A mudança de fase da água para que essa transferência ocorra, consome energia e, conseqüentemente, causa a diminuição da temperatura do ar úmido. Esse fato explica a eficiência do sistema evaporativo da Equação 9.

O outro fator que influencia na velocidade de reação de transferência de umidade para o ar é a superfície de contato entre a água e o ar. Quanto maior a superfície de contato, maior será a velocidade de transferência. O aumento da superfície de contato é feito pelo uso de dispersores de água.

Segundo os tipos de dispersores de água, os sistemas evaporativos de refrigeração podem ser divididos em duas categorias diferentes [10]:

1. Resfriamento Evaporativo Direto (RED)
2. Resfriamento Evaporativo Indireto (REI)

Nos sistemas de resfriamento evaporativos diretos, Figura 3-a, o ar é resfriado por meio do contato direto com a superfície líquida. Nos sistemas de resfriamento evaporativos indiretos, Figura 3-b, existe uma separação entre dois fluxos diferentes de ar, onde um está em contato direto com a superfície líquida e é resfriado e o outro é resfriado por condução. Esse segundo fluxo de ar é mantido praticamente seco.

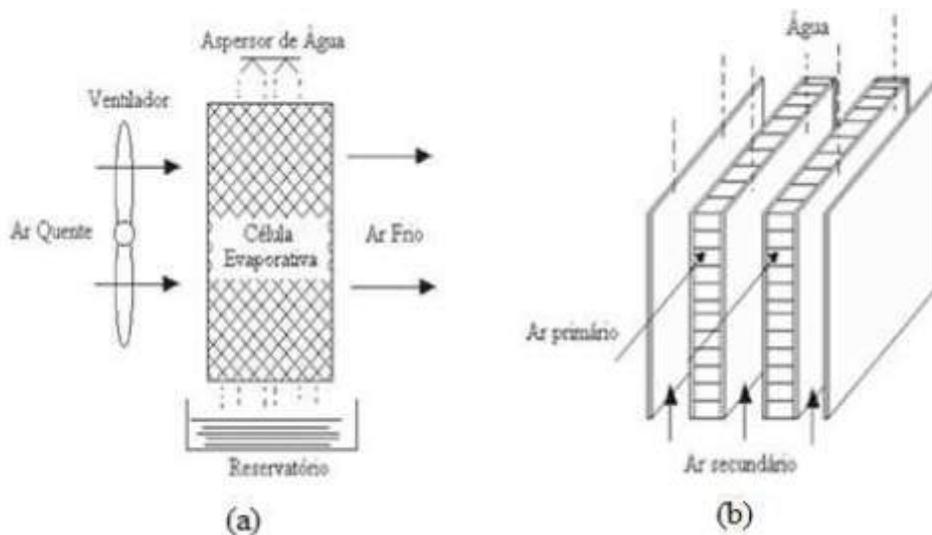


Figura 3. (a) Resfriamento evaporativo Direto. (b) Resfriamento evaporativo Indireto

Uma característica que deve ser observada no sistema de resfriamento evaporativo do ar é que a temperatura de insuflação do ar depende da temperatura do ar de entrada e da umidade relativa do ar de entrada, ou seja, depende da temperatura de bulbo úmido do ar externo, como dita a Equação (3). Dessa forma, nem sempre é possível atingir-se uma temperatura de conforto térmico. A Tabela 2 demonstra essa relação entre a temperatura do ar de entrada e sua umidade relativa.

Tabela 2. Redução da temperatura interna em função da umidade relativa do ar externo e sua temperatura [16]

Umidade Relativa do ar externo	Temperatura do ar externo				
	25°C	30°C	35°C	40°C	45°C
	Redução de Temperatura/Temperatura Final (°C)				
30%	8,5 / 16,5	9,6 / 20,4	10,8 / 24,2	12,0 / 28,0	15,0 / 30,0
40%	7,1 / 17,9	8,0 / 22,0	8,8 / 26,2	9,7 / 30,3	12,5 / 32,5
50%	5,7 / 19,3	6,5 / 23,5	7,1 / 27,9	7,8 / 32,2	9,5 / 35,5
60%	4,4 / 20,6	5,0 / 25,0	5,5 / 29,5	5,9 / 34,1	7,0 / 38,0
70%	3,2 / 21,8	3,6 / 26,4	4,0 / 31,0	4,4 / 35,6	5,0 / 40,0

A norma NR 17[20] dita que, para o conforto térmico, a temperatura do ambiente de trabalho onde são executadas atividades intelectuais deve ficar em torno de 23°C, com umidade relativa entre 40% e 80%. Conforme a Tabela 2, temperaturas externas acima dos 30°C com umidade relativa acima dos 60% torna o resfriamento evaporativo do ar insuficiente para garantir o conforto térmico.

2.5 SISTEMA CONVENCIONAL DE REFRIGERAÇÃO

Refrigeração pode ser definida como um processo de retirada e transferência de calor, isto é, retira-se calor indesejado de um espaço ou corpo para reduzir sua temperatura. Essa temperatura retirada é então transferida mecanicamente para outro espaço ou corpo.

Um sistema convencional de refrigeração usual é o *split*, que é um sistema dividido em dois módulos, denominados unidade interna (evaporadora) e unidade externa (condensadora). A unidade evaporadora fica dentro do ambiente que será resfriado e a unidade condensadora fica localizada em um ambiente externo. A união entre esses dois módulos é feita através de tubos de alta e baixa pressão, que levam, respectivamente, o fluido refrigerante no estado de vapor para o condensador e o fluido refrigerante líquido para o evaporador.

O líquido refrigerante possui características termodinâmicas especiais, podendo vaporizar-se a temperaturas muito baixas quando comparadas à temperatura de vaporização da água. O seu ponto de ebulição é inferior à temperatura ambiente. Na condensação do fluido refrigerante acontece a rejeição de calor e, na sua evaporação, acontece a absorção de calor. Esse líquido é circulado pelo sistema e é transformado periodicamente em líquido e vapor, promovendo a remoção de calor de uma fonte quente [11].

Uma das características dos sistemas convencionais de refrigeração é que quando o ar quente e úmido do ambiente a ser refrigerado entra em contato com a superfície fria do tubo aletado da unidade evaporadora, sofre o processo de condensação, porque muitas das vezes essas aletas estão na temperatura de ponto de orvalho. Esse processo acaba por diminuir a umidade relativa do ar do ambiente.

2.6 SISTEMA DE CLIMATIZAÇÃO HÍBRIDA

O sistema de climatização híbrida é composto por dois sistemas de refrigeração: sistema convencional de refrigeração e sistema de resfriamento evaporativo do ar [1].

O sistema convencional de refrigeração possui maior capacidade de diminuição da temperatura do ar se comparado ao sistema de resfriamento evaporativo. Além disso, enquanto o sistema convencional diminui a umidade do ar por condensação da água, o sistema evaporativo a aumenta. O sistema convencional consome mais energia elétrica do que o sistema evaporativo. Então, visando a eficiência energética, o objetivo do sistema de climatização híbrida é utilizar as vantagens de cada um dos sistemas, privilegiando-se o uso do sistema de resfriamento evaporativo, uma vez que esse consome menos energia elétrica.¹

2.7 SISTEMA SUPERVISÓRIO OPENHAB

O openHAB, acrônimo de *open Home Automation Bus*, é um software livre de automação que visa a integração de diferentes sistemas e tecnologias em uma única solução, que permite uma implementação abrangente de regras de automação e o uso de interface gráfica [13].

Escrito inteiramente em na linguagem Java e com arquitetura de modularização OSGi, o openHAB foi projetado para ser executado em uma variedade de sistemas operacionais (Windows, Linux, OS X) e configurações de hardware.

O openHAB é dito ser agnóstico de hardware e protocolo. Como agnóstico de protocolo, o openHAB é capaz de estabelecer comunicação com qualquer protocolo que o usuário escolher, permitindo uma integração de uma grande variedade de dispositivos.

O openHAB pode ser controlado por uma única interface gráfica acessível de qualquer navegador padrão ou aplicativos de celular.

2.7.1 ARQUITETURA

O sistema do openHAB é dividido em duas partes:

- openHAB Runtime
- openHAB Designer

2.7.1.1 OPENHAB RUNTIME

Pode-se definir um sistema *runtime* como um conjunto de recursos de hardware e software que habilitam um programa a ser executado em um computador. Tipicamente, um sistema runtime fornece comandos de baixo e alto nível de um modelo de execução de uma linguagem de programação, fazendo a interface entre a estrutura do software (*framework*) e o conjunto de instruções da arquitetura de hardware (ISA) [14].

¹ A Tabela 9 mostra um comparativo do consumo de cada parte do sistema

As funções de baixo nível fornecidas pelo sistema *runtime* incluem, por exemplo, o carregamento da memória, execução paralela de softwares, a conversão digital para binário, a correta manipulação da pilha do processador após uma chamada de função, entre outras.

O openHAB Runtime é um conjunto de pacotes de estrutura OSGi, sendo assim, é uma solução Java que requer uma Máquina Virtual Java (JVM) para ser executado. A estrutura OSGi é uma arquitetura altamente modular que permite que o usuário adicione ou remova diversas funcionalidades durante sua execução, sem a necessidade de parar o serviço.

A comunicação é tratada de duas formas diferentes pelos canais internos do openHAB: um barramento assíncrono de eventos e um repositório de estados.

O serviço básico do openHAB é o Event Bus (barramento assíncrono) que trata de dois tipos diferentes de eventos:

- Comandos que disparam uma ação ou mudança de estado de um item ou serviço
- Atualizações que fornecem informações sobre o estado de mudança de um item ou serviço.

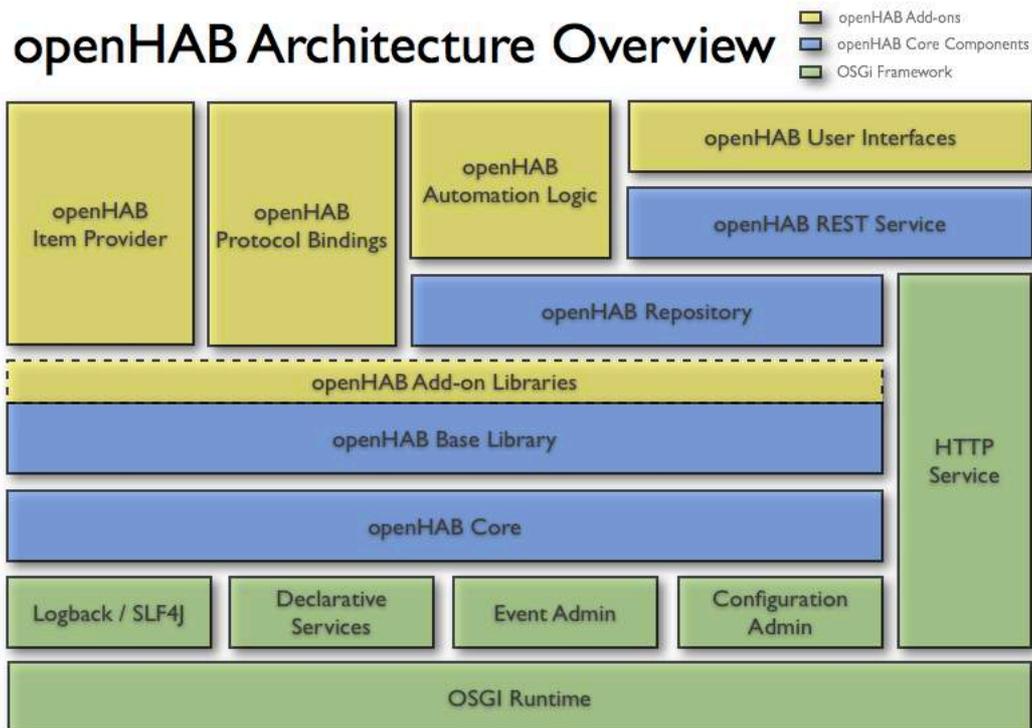


Figura 4. Arquitetura do openHAB

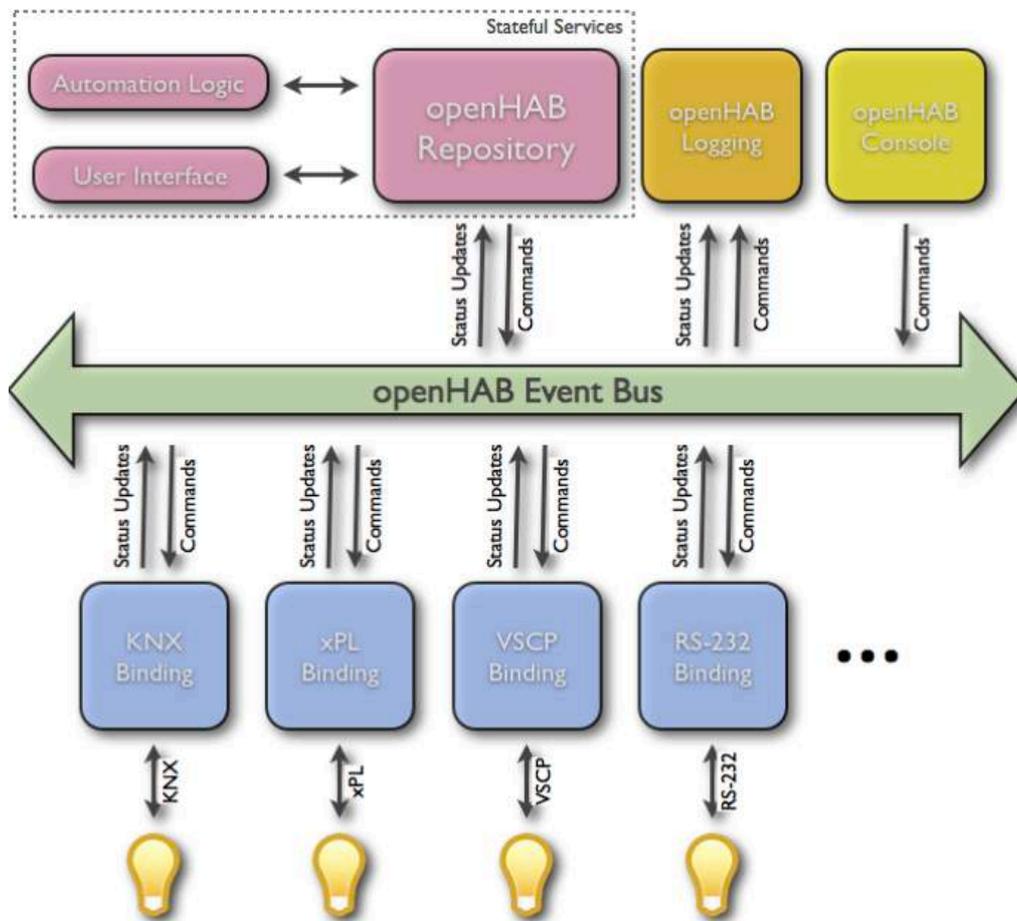


Figura 5. Barramento de eventos do openHAB

O openHAB serve como um hub de integração que se coloca entre o hardware dos dispositivos e atua como um mediador entre os diferentes protocolos implementados por esses dispositivos. Dessa forma, apenas a instância do openHAB necessita estar rodando em um servidor central para facilitar a comunicação entre os diferentes dispositivos.

Conectado ao barramento de eventos do openHAB está o repositório de estados do openHAB, que mantém o controle o estado atual de todos os itens. O repositório de estados é usado sempre que for necessário acessar o estado atual de um item que foi solicitado pela interface gráfica ou para manter a lógica de execução informada dos estados de todos os itens.

A interface gráfica do openHAB é construída em torno de uma configuração genérica chamada de Sitemap. O Sitemap é uma estrutura de árvore de mecanismos projetados para definir diferentes páginas da interface gráfica do usuário e todo o seu conteúdo.

2.7.1.2 OPENHAB DESIGNER

O openHAB designer é uma aplicação *Eclipse Rich Client Platform* (RCP) para configurar o openHAB Runtime [13].

Uma aplicação Eclipse é um ambiente de desenvolvimento Java de software livre. É comumente usado para desenvolvimento de linguagem Java, mas suporta muitas outras linguagens, como a implementação do openHAB, que tem uma linguagem própria.

Uma Plataforma de Cliente Rica ou *Rich Client Platform* é uma plataforma de desenvolvimento que contém os seguintes componentes:

- Um gerenciador de ciclo de vida principal (*microkernel*)
- Uma plataforma padrão de agregação
- Um kit de ferramentas de desenvolvimento
- Buffers de arquivos, manipulação de texto e editores de texto
- Facilidades de ligação de dados (data binding), que é uma técnica que une duas fontes de dados e as mantém em sincronia em um processo que estabelece uma conexão entre a interface de usuário da aplicação e a lógica de controle do serviço
- Um gerenciador de atualizações

Dessa forma, o openHAB Designer inclui editores para configurar os Sitmaps, definindo-se itens ou grupo de itens, assim como escrever as regras que governarão o comportamento desses itens ou grupos de itens. Fornece também um editor de texto que oferece a verificação de sintaxe, preenchimento automático de texto, realce de texto e um assistente de conteúdo, que aponta possíveis erros de linguagem.

A interface do openHAB Designer é mostrada na Figura 6.

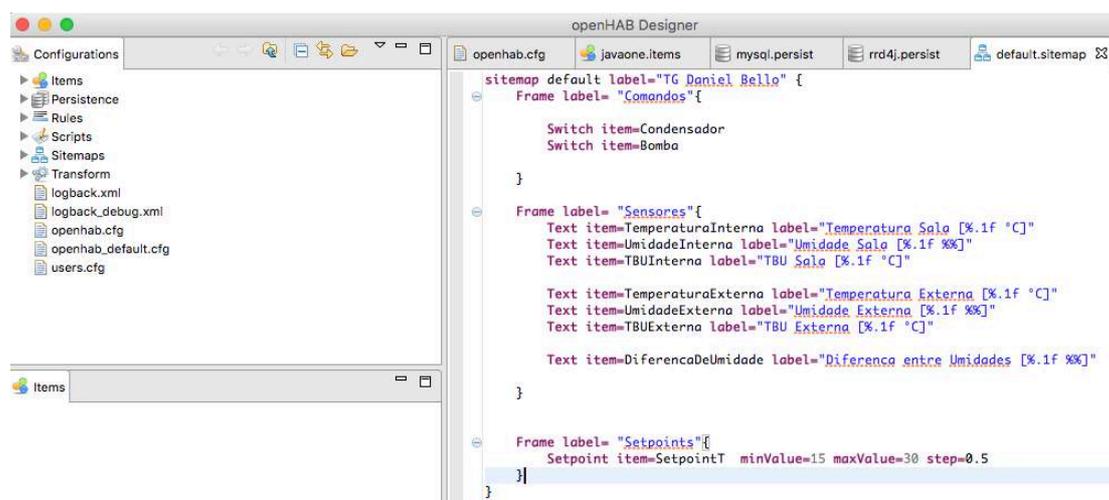


Figura 6. openHAB Designer

2.7.2 LIGAÇÕES (BIDINGS)

Ligações são pacotes adicionais que podem ser usados para estender as funcionalidades do openHAB. Essas ligações são possíveis devido ao fato de o openHAB utilizar o *middleware* OSGi. Por ser um software livre, o número de dispositivos e protocolos que podem ser adicionados pelas ligações cresce diariamente.

Alguns exemplos de dispositivos e protocolos que podem ser ligados são:

- ModBus TCP
- HTTP

- MQTT
- Asterisk
- Open Energy Monitor
- Z-Wave
- Serial
- SNTP
- Termostato Nest

2.7.3 BANCO DE DADOS

Banco de dados é um conjunto de arquivos relacionados entre si com registros sobre pessoas, lugares ou informações em geral. São coleções organizadas de dados que se relacionam de forma a criar informações e dar mais eficiência durante uma pesquisa ou estudo [15].

Em sistemas de automação, existe a necessidade de se guardar os dados de diferentes equipamentos e sensores sobre determinado tempo. Em regra, a gravação desses dados ocorre em intervalos fixos de tempo, na mudança de um valor ou disparado por uma variável do sistema.

No openHAB, o bando de dados é adicionado pelas Ligações (*bindings*). Os diferentes tipos de serviço de banco de dados que funcionam integrados com o openHAB são:

- Cosm
- Db4o
- InfluxDB
- JDBC
- Logging
- MySQL
- RRD4J
- Sen.se

O usuário tem a possibilidade de escolher quantos serviços de banco de dados quiser e adicioná-los com as Ligações do openHAB.

2.7.4 REGRAS

O openHAB possui um mecanismo de regras para automação incluído em seu núcleo. Essas regras são escritas em uma linguagem própria do openHAB similar com Java. Tais regras podem ser disparadas e utilizadas por diferentes tipos de eventos, como mudanças em um estado de dispositivo, tempo ou eventos do sistema.

Além de definir lógica flexíveis, as regras do openHAB fornecem um conjunto de diferentes ações que podem ser tomadas por essas regras, como enviar comandos e atualizações para dispositivos.

2.8 PROTOCOLO DE COMUNICAÇÃO MQTT

Acrônimo de *Message Queue Telemetry Transport*, o MQTT é um protocolo aberto e padrão ISO/IEC PRF 20922 leve de mensagens para sensores (telemetria) e pequenos dispositivos móveis otimizado para redes TCP/IP não confiáveis ou de alta latência. O esquema de troca de mensagens é fundamentado no modelo Publicador-Subscritor [12].

O modelo Publicador-Subscritor define uma dependência um-para-muitos entre objetos de modo que, quando um objeto muda o estado, todos os seus dependentes são notificados e atualizados automaticamente. Assim, permite que objetos interessados sejam avisados da mudança de estado ou outros eventos que estão ocorrendo em outro objeto. Para ser avisado, um objeto deve estar subscrito em um tópico, que é o canal utilizado para a troca de mensagens.

Esse modelo Publicador-Subscritor do MQTT requer um corretor de mensagens ou *message broker*. Um *message broker* é um programa/módulo intermediário que traduz a mensagem em um dado protocolo do remetente para uma mensagem no protocolo do destinatário.

O MQTT tem métodos definidos para indicar a ação a ser tomada por um recurso identificado. Alguns métodos definidos são:

- *Connect*: aguarda por uma conexão a ser estabelecida com o servidor/broker
- *Disconnect*: aguarda um cliente MQTT terminar sua ação e desconectar a sessão TCP/IP
- *Subscribe*: realiza a subscrição de um cliente em determinado tópico
- *Unsubscribe*: termina a subscrição de um cliente em determinado tópico
- *Publish*: publica uma mensagem em determinado tópico. Todos os subscritos nesse tópico estão aptos a receber a mensagem.

A Tabela 3 mostra uma comparação entre o protocolo MQTT e o protocolo HTTP.

Tabela 3. Comparação entre MQTT e HTTP[12]

	MQTT	HTTP
Orientação de Projeto	Centralizado em Dados	Centralizado em documentos
Padrão	Publicação/Subscrição	Pedido/Resposta
Complexidade	Simple	Complexo
Tamanho da mensagem	Pequena, com um cabeçalho binário compacto de 2 bytes	Grande, em parte devido ao detalhe de estado ser baseado em texto
Nível de serviços (QoS)	Três níveis de qualidades de serviços	Todos as mensagens tem o mesmo nível de qualidade
Distribuição de dados	Suporta 1 para 0, 1 para 1 e 1 para n	Apenas 1 para 1

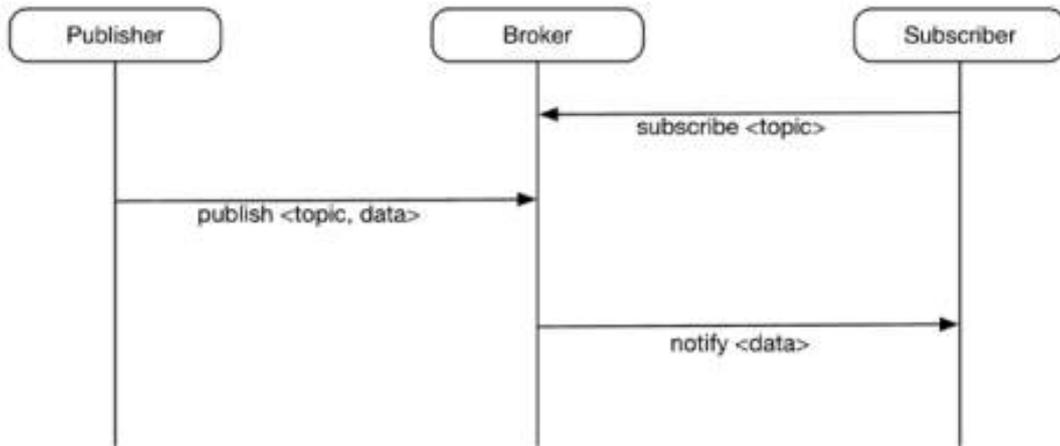


Figura 7. Arquitetura do paradigma publish/subscribe [12]

O paradigma publish/subscribe (Figura 7) utiliza o conceito de tópicos para processar as mensagens, em que cada mensagem é enviada para um determinado tópico. Diferentemente de outros protocolos de mensagem, o *publisher* não envia a mensagem diretamente ao *subscriber*, mas sim ao *broker*. Essa mensagem é enviada ao *broker* em um determinado tópico. O *broker* é responsável por receber a mensagem do *publisher* e fazer uma pré-filtragem dessas mensagens e enviá-las aos *subscribers* que estiverem registrados em um determinado tópico.

Os tópicos são multiníveis, ou seja, podem ter 1 ou “n” níveis. Como exemplo, pode-se supor que a temperatura da sala de uma determinada casa esteja sendo medida. Pode-se criar um nível mestre “casa”, seguido por “sala” e então de “temperatura”. O resultado será casa/sala/temperatura. O cliente pode inscrever-se para o tópico exato ao qual se deseja registrar dados ou usar um coringa para outro nível. Inscrever-se para casa/+/temperatura enviará a temperatura para todos os tópicos do segundo nível como por exemplo, casa/cozinha/temperatura. Por fim, um coringa multi-nível é a cerquilha (#). Isso permite inscrever-se para todos os níveis hierárquicos. Dessa forma, casa/# servirá para todos os níveis contidos após casa.

2.8.1 ECLIPSE MOSQUITTO BROKER

O *broker* (corretor) de mensagens Eclipse Mosquitto é um projeto software livre escrito em linguagem C que implementa uma versão pequena do protocolo de mensagens MQTT [12]. Pequeno significa que:

- Apenas as funções necessárias foram incluídas – uma compilação condicional pode ser usada para habilitar funções omitidas
- As funções são codificadas de forma mais eficientes e otimizadas o possível
- As funções para enviar mensagens são implementadas da forma mais simples possível para que o protocolo funcione.

A razão do Mosquitto ser implementado na linguagem C é para que o servidor possa rodar em máquinas que não podem rodar Máquinas Virtuais Java (JVM). Dessa forma, o Mosquitto pode implementar o protocolo de comunicação MQTT mesmo em máquinas com sistema embarcado, usando o mínimo de energia e memória.

Tipicamente, uma implementação do Mosquitto pode ser executada consumindo 120 kB de memória e 3MB de RAM, com 100 clientes conectados.

2.9 CARGA ELÉTRICA E CONSUMO DE ENERGIA ELÉTRICA

Pode-se definir carga elétrica como um conjunto de valores de grandezas físicas que caracterizam as solicitações impostas em dado instante a um sistema ou equipamento elétrico ou a um componente a ele ligado, seja elétrico ou não. A carga pode ser expressa em termos de impedância, de corrente, de potência ativa, reativa ou aparente, ou de uma característica não elétrica, conforme as circunstâncias peculiares a cada caso.

Assim, pode-se dizer que uma carga é que qualquer equipamento ou conjunto de equipamentos ligados a um sistema elétrico e absorvendo potência desse sistema.

O consumo de energia elétrica mensura a quantidade de energia consumida por uma carga elétrica durante um certo período de tempo. É mensurado em watt-horas ($W \cdot h$).

$$1W \cdot h = 3600 \text{ joules} = 859,8 \text{ calorias} \quad (10)$$

O consumo de energia elétrica por ser calculado pela Equação (11) [21]

$$E = Pot \cdot \Delta t \quad (11)$$

onde

E = Consumo de energia elétrica, [$W \cdot h$]

Pot = Potência Ativa da Carga, [W]

Δt = Tempo de consumo, [h]

CAPÍTULO 3 – SISTEMA HÍBRIDO DE CLIMATIZAÇÃO DA SALA DE REUNIÕES DO LARA

3.1 ASPECTOS CONSTRUTIVOS

O sistema híbrido de climatização da sala de reuniões do LARA foi construído com um sistema de resfriamento evaporativo do ar em série com um sistema de refrigeração convencional do tipo *split*. A Figura 8 mostra o projeto executado.

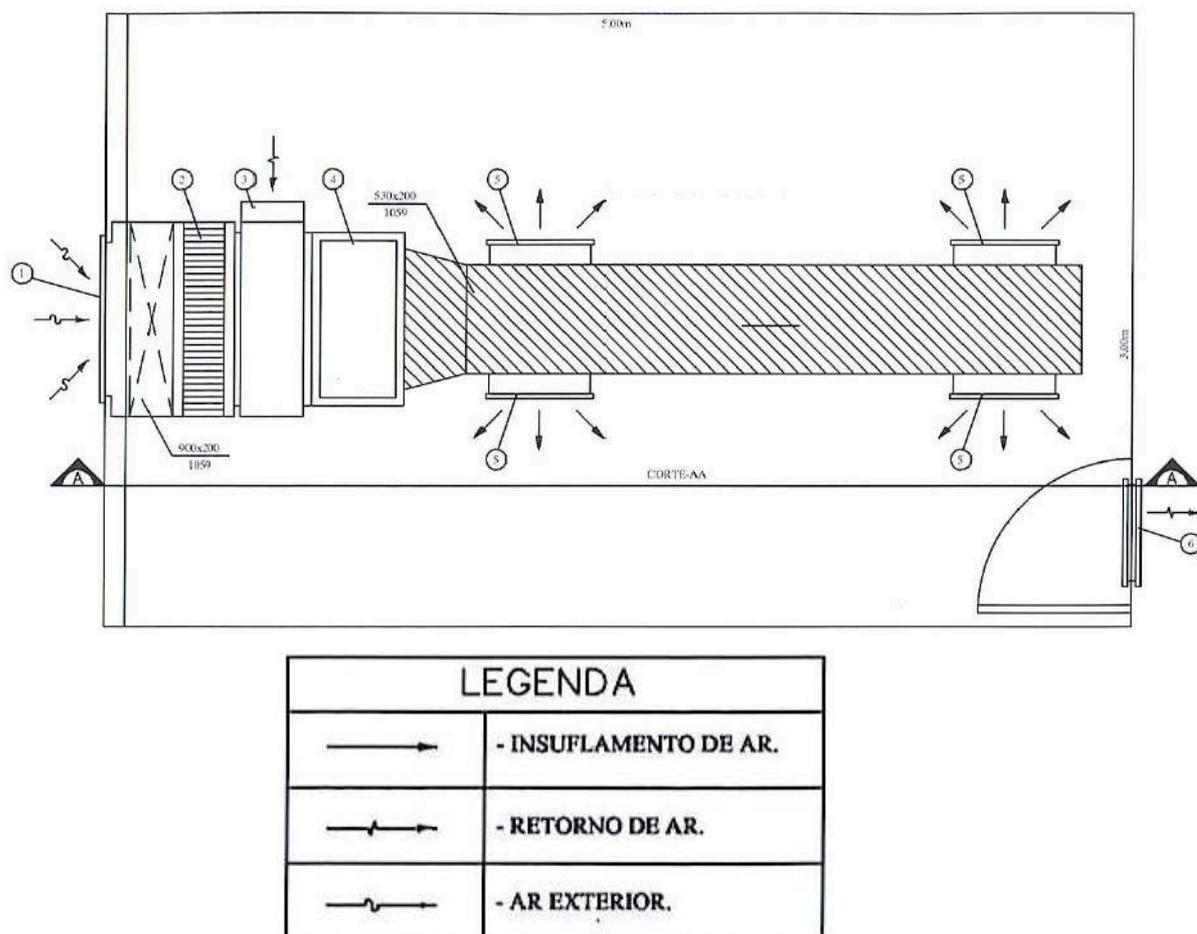


Figura 8. Projeto do Sistema Híbrido de Climatização da Sala de Reuniões do LARA [1]

Tabela 4. Lista de materiais utilizados na implementação do sistema de ar condicionado híbrido da sala de reuniões do LARA, juntamente com a legenda dos itens apontados na Figura 8 [1]

Lista de Materiais		
Item	Descrição	Quantidade
1	Veneziana ar exterior, Trox	1
2	Resfriador evaporativo	1
3	Damper para motorização, Trox	1
4	Slipt System Versatile Dutável, Carrier – 18.000 BTUs	1
5	Grelha de insuflamento, Trox	4
6	Grelha de retorno, Trox	1

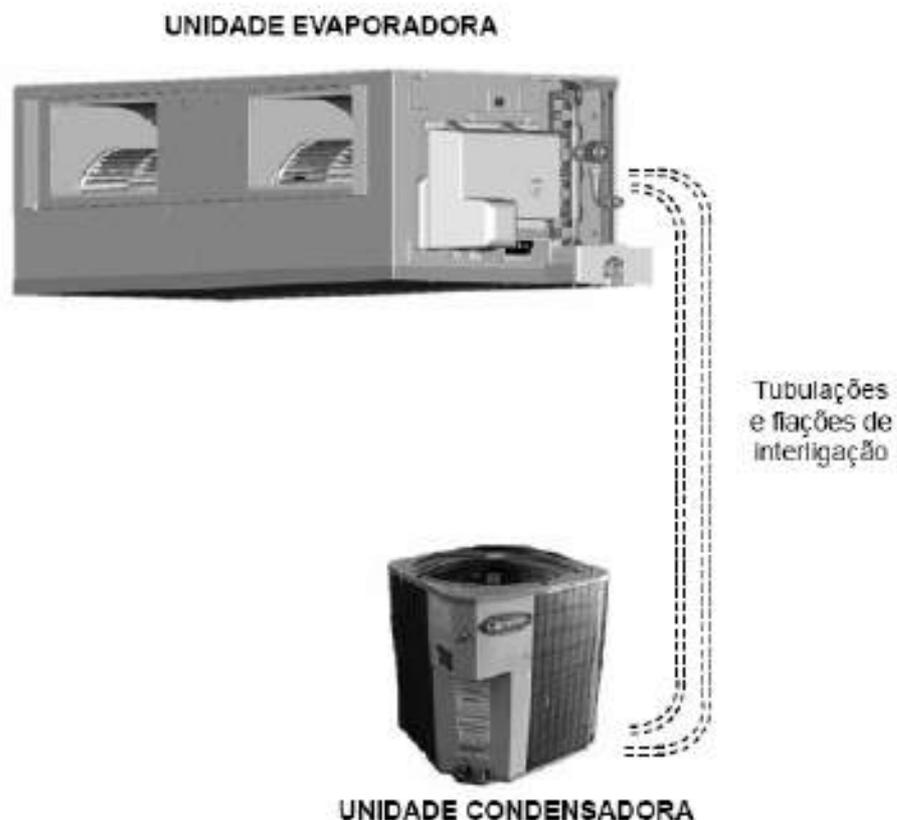


Figura 9. Sistema de refrigeração *split* Versatile dutável da Carrier – 18.000 BTUs [17]

A unidade evaporadora possui internamente um ventilador radial de três velocidades que pode ser acionado separadamente. Esse ventilador é o responsável por insuflar o ar pelo duto de ar.



Figura 10. Dispensador de água de celulose da Munters usado no sistema evaporativo [16]

A bomba d'Água , Figura 11, está instalada dentro de um reservatório de água, sendo a responsável por jogar água em cima da manta de celulose, para que essa seja dispersa no ar. A bomba d'Água juntamente com a manta de celulose formam a base do sistema de climatização por resfriamento evaporativo direto (RED).



Figura 11. Bomba d'Água Átima 8W para dispersão de água na manta de celulose.

O atuador do *dampers*, Figura 12, funciona com tensão contínua e varia sua abertura proporcionalmente à tensão de entrada (0~10Vcc). Dessa forma, a tensão aplicada nos seus terminais para que determinado percentual de abertura seja realizado é dada pela Equação 12.

$$T_{ENTRADA} = 10 \times \%Abertura \quad (12)$$

Assim, para uma abertura de 30%, a tensão que deve ser aplicada nos terminais do atuador do *dampers* deve ser igual a 3Vcc.



Figura 12. Atuador para *damper*, modelo AM-2001, 24 Vac x 0 ~ 10Vcc

A placa eletrônica DRT-3403-N, Figura 13, implementa o controle de abertura e fechamento proporcional do atuador do *damper* através da leitura de um sensor de temperatura ou através de um potenciômetro. Para isso, a placa, juntamente com um retificador, transforma uma entrada 24 Vac em uma tensão de saída variando entre 0 e 10 Vcc.

Nesse projeto, o potenciômetro da placa foi configurado para que a tensão de saída seja sempre 10 Vcc. Dessa forma, o sistema de automação implementado aciona a placa para realizar a abertura total do *damper*. Para o fechamento total do *damper*, o sistema de automação deixa de acionar a placa.



Figura 13. Placa eletrônica DRT-3403-N, 24 Vac x 0 ~ 10Vcc [18]

3.2 QUADRO DE COMANDO

A Figura 14 mostra o diagrama de blocos unifilar para o Sistema Híbrido de Climatização da Sala de Reuniões do LARA. A descrição do seu funcionamento será apresentada na seção 3.6.

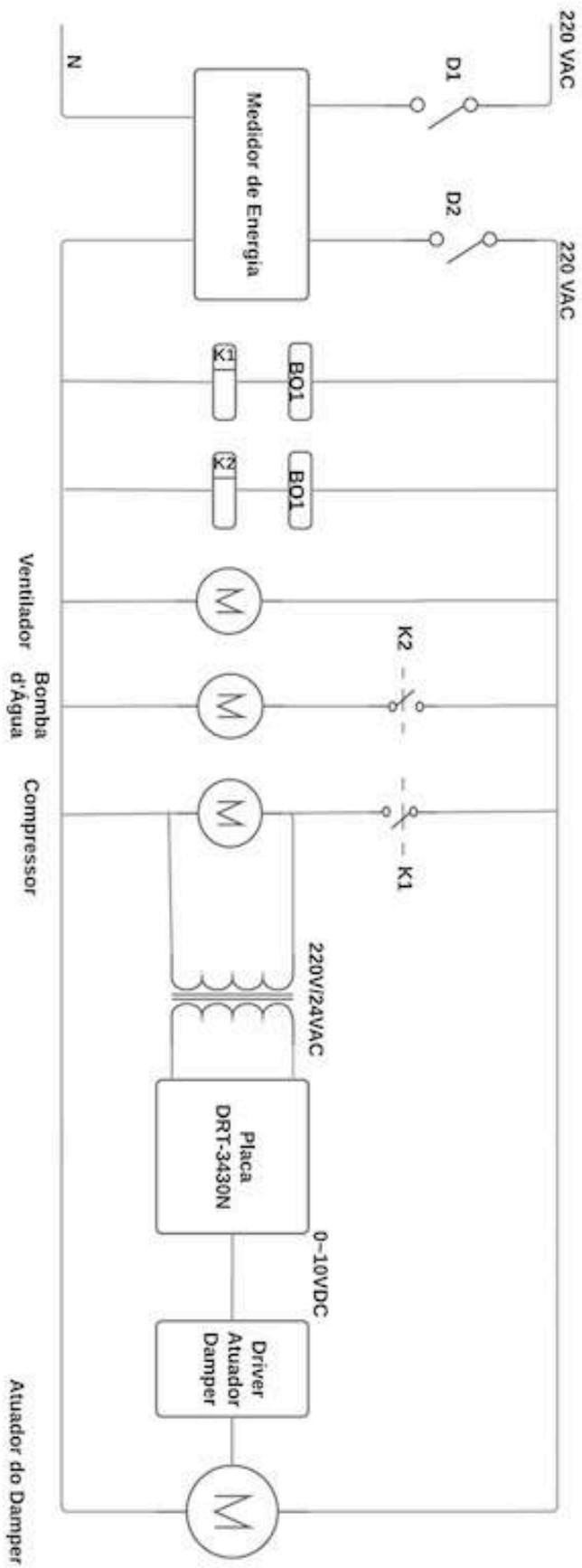


Figura 14. Diagrama de blocos unifilar para o Sistema Híbrido de Climatização da Sala de Reuniões do LARA

3.3 SENSORIAMENTO

Essa seção apresenta os sensores utilizados para mensurar e controlar a temperatura e a umidade da sala de reuniões. Para esse objetivo, foram utilizados dois sensores DHT22, sendo que um está localizado dentro da sala de reuniões e o outro na parte externa, protegido contra radiação.

3.3.1 SENSOR DE TEMPERATURA E UMIDADE DHT22

O sensor DHT22 é um sensor digital de temperatura e umidade calibrado. Os seus elementos sensores estão ligados a um microcontrolador próprio de 8 bits.

Cada sensor desse modelo tem sua temperatura compensada e calibrada em uma câmara de calibração e seu coeficiente de calibração é salvo em uma memória OTP (*One-Time Programmable*). Toda vez que o sensor é lido, ele irá ler esse coeficiente de sua memória.

Tem como característica o tamanho pequeno, o baixo consumo e boa distância de transmissão de dados (20m).



Figura 15. Sensor de temperatura e umidade DHT22

Esse sensor possui 4 pinos, porém um deles não é utilizado. A descrição da função dos seus pinos encontra-se na Tabela 5.

Tabela 5. Pinos do sensor DHT22

Pino	Função
1	VDD – Tensão de Alimentação
2	DATA – Serial data, bidirecional
3	NULL – Não utilizado
4	GND – Ground (referência)

Tabela 6. Especificações técnicas do sensor DHT22

Modelo	DHT22	
Tensão de alimentação	3,3-6Vcc	
Sinal de saída	Sinal digital via barramento simples	
Elemento Sensor	Capacitor de polímero	
Faixa de Operação	Umidade: 0-100%	Temperatura: -40~80 °C
Acurácia	Umidade: $\pm 2\%$	Temperatura: $< \pm 5^{\circ}\text{C}$
Resolução ou sensibilidade	Umidade: 0,1%	Temperatura: 0,1°C
Repetibilidade	Umidade: $\pm 1\%$	Temperatura: $\pm 0,2^{\circ}\text{C}$
Histerese de umidade	$\pm 0,3\%$	
Estabilidade a longo prazo	$\pm 0,5\%$ /ano	
Período de sensoriamento	Em média: 2 segundos	
Permutabilidade	Totalmente permutavel	
Dimensões	Vista Lateral: 14x18x5,5mm; Vista Frontal: 22x28x5mm	

3.3.2 CALIBRAÇÃO DO SENSOR DHT22

Como foi dito na seção anterior, o sensor DHT22 vem calibrado de fábrica, onde o seu coeficiente de calibração é salvo em uma memória OTP. Para efeito de comparação, utilizou-se o equipamento 3M Quest Temp° 36[®].

3M Quest Temp° 36[®] realiza medições simultâneas de temperatura, umidade e cálculo de temperatura efetiva. Tem como características:

- Certificação de segurança
- Medições:
 - Temperatura de bulbo seco: $\pm 0,5$ de 0°C a 120°C
 - Temperatura de bulbo úmido: $\pm 0,5$ de 0°C a 120°C
 - Umidade Relativa: $\pm 5\%$ de 20 a 95%
- Temperatura de Operação: -5°C a 100°C
- Proteção: IP-54
- Conformidade eletromagnética: CE Mark
- Normas Atendidas:
 - ISO 7243
 - NHO 06

- NR15
- NR9
- Norma UL/CSA para Classe I, Grupos A, B, C e D
- Classe II Grupos E, F e G
- Classe III Código de temperatura T3&ATEX.

Comparando-se as leituras de temperatura e umidade dos sensores DHT22 com o 3M Quest Temp^o 36^o, verifica-se que a diferença entre as temperaturas ficaram em $\pm 0,1^{\circ}\text{C}$ e as diferenças das umidades em $\pm 0,1\%$. Assim, para efeitos de conforto térmico, pode-se assumir que as leituras estão calibradas.



Figura 16. Estação de monitoramento de temperatura e umidade 3M Quest Temp^o 36^o.

3.4 VISÃO GERAL DO SISTEMA DE AUTOMAÇÃO

Os equipamentos utilizados para a implementação do sistema de automação projetado para controlar o Sistema Híbrido de Climatização estão descritos na Tabela 7. A Figura 17 mostra um diagrama geral do sistema de automação implementado.

Tabela 7. Lista de equipamentos do sistema de automação

Item	Descrição	Quantidade
1	Raspberry Pi 2 Model B	1
2	Arduino Mega 2560	1
3	Módulo Ethernet ENC 28J60	1
4	Módulo Relé 5V com 4 relés	1
5	Roteador sem fio D-Link com 4 portas	1
6	Sensor de Temperatura e Umidade DHT22	2

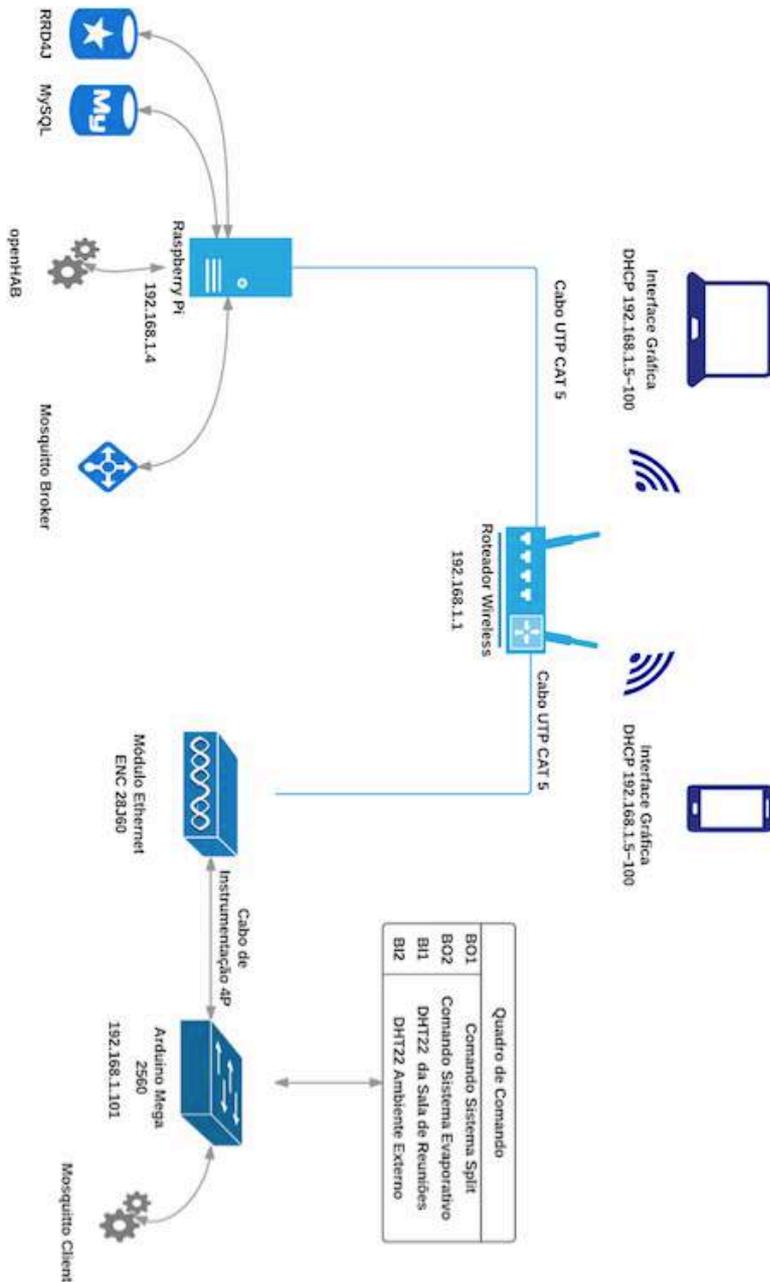


Figura 17. Diagrama Geral do Sistema de Automação

3.4.1 RASBERRY PI

O Raspberry Pi é um computador no tamanho de um cartão de crédito baseado em um System-on-a-Chip (SoC). O modelo escolhido para esse projeto foi o Raspberry Pi 2 Model B, que apresenta as seguintes especificações:

- Chip: Broadcom BCM2835 SoC
- Processador: ARM Quad-Core Cortex-7
- CPU: 900 MHz
- GPU: Broadcom VideoCore IV

- Tensão de operação: Micro USB Socket 5V/2A
- 4 portas USB
- 40 pinos de GPIO
- Full HDMI
- Porta Ethernet 10/100 (RJ45)
- Slot Micro SD Card

O sistema operacional instalado no Raspberry Pi foi o Raspbian Jessie 4.4, que é uma variante do kernel Linux Debian. Nele estão instalados:

- OpenHAB Runtime, com os seus respectivos arquivos de configuração
- Banco de dados MySQL
- Banco de dados RRD4J
- Mosquitto Broker
- Máquina Virtual Java (JVM)
- Servidor de Arquivos Samba, para auxiliar na troca de arquivos de configuração pela rede

Nesse projeto, o Raspberry Pi está definido com IP fixo 192.168.1.4.

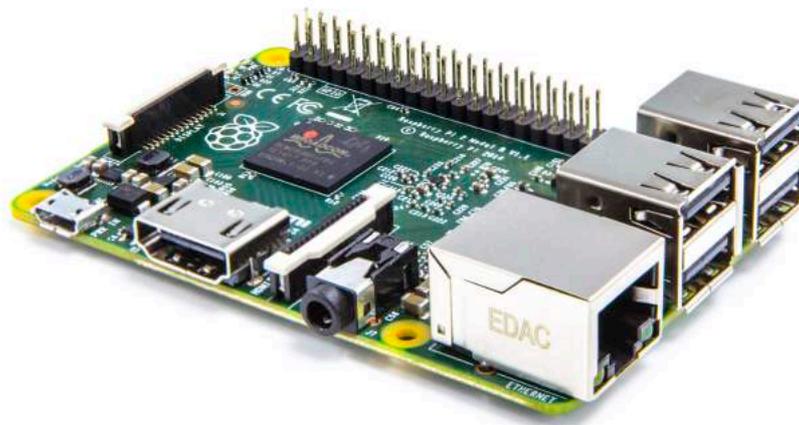


Figura 18. Raspberry Pi 2 Model B

3.4.2 ARDUINO

O Arduino é uma placa de prototipagem eletrônica de hardware livre e de placa única, projetada com um microcontrolador Atmel AVR com suporte de entrada e saída embutido, uma linguagem de programação padrão, que tem origem na linguagem Wiring, que é uma modificação das linguagens C e C++.

Entre as diferentes placas de Arduino existentes no mercado, para este projeto a escolhida foi a Mega 2560, por apresentar maior capacidade de processamento se comparado com as demais placas, além do seu grande número de entradas e saídas. Essa placa é baseada no microcontrolador ATmega 2560. Suas especificações técnicas estão expostas na Tabela 8.

Tabela 8. Especificações técnicas do Arduino Mega 2560

Microcontrolador	Atmega 2560
Tensão de Operação	5Vcc
Tensão de entrada (recomendado)	7-12Vcc
Tensão de entrada (limite)	6-20Vcc
Pinos de entrada e saída digitais	54, sendo que 15 desses podem fornecer saída PWM
Pinos de entrada analógica	16
Corrente DC por pino de entrada e saída	20 mA
Memória Flash	256 KB, sendo que 8KB é usado pelo bootloader
SRAM	8 KB
EEPROM	4 KB
Velocidade de Clock	16 MHz
Peso	37g



Figura 19. Arduino Mega 2560

Neste projeto, o Arduino foi usado como dispositivo de entrada e saída de dados e interface de comandos. Ele é o responsável por receber os dados de leitura dos sensores DHT22, processá-los e enviá-los ao openHAB via protocolo MQTT. Os comando para ligar e desligar o sistema evaporativo e o sistema de refrigeração convencional são enviados pelo openHAB ao Arduino, que por sua vez faz os acionamentos das cargas usando o módulo relé.

Os sensores enviam ao Arduino dados de temperatura de bulbo seco (TBS) e umidade relativa do ar. Usando-se a Equação 3, o Arduino faz o cálculo da temperatura de bulbo úmido (TBU).

3.4.3 MÓDULO ETHERNET ENC 28J60

O Arduino não possui uma porta ethernet. Por isso foi necessário adicionar-se esse driver para que o Arduino pudesse conectar-se ao Mosquitto Broker. Dessa forma foi

possível integrar esse sistema com o software openHAB, embarcado no Raspberry Pi.

O driver utiliza o chip da Microchip® ENC 28J60, que é um controlador de internet IC que suporta diversos protocolos de rede.

Com o uso desse driver, o Arduino recebeu o endereço de IP fixo 192.168.1.101.



Figura 20. Módulo Ethernet ENC 28J60

3.4.4 MÓDULO RELÉ

Um relé pode ser definido como um interruptor eletromecânico. A movimentação física desse interruptor ocorre quando uma corrente elétrica percorre as bobinas de suas espiras, criando um campo magnético que desloca uma alavanca responsável pela mudança de estados de seus contatos. A Figura 21 mostra o desenho esquemático de um relé [21].

Uma característica importante do relé é que o circuito controlado pode ter características totalmente distintas do circuito que controla a bobina. Assim, se a bobina for feita com muitas espiras de fio muito fino, como ocorre com micro relés, uma corrente fraca, como a que obtemos de um transistor, ou mesmo diretamente de um sensor, pode fechar os contatos do relé, e com isso controlar aparelhos que exigem correntes maiores como, por exemplo, lâmpadas, motores, entre outros.

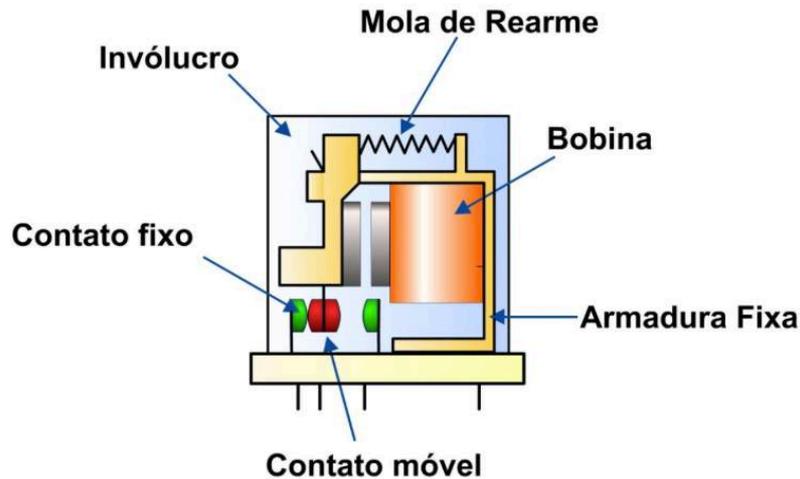


Figura 21. Desenho esquemático de um relé

Neste projeto, os relés são acionados por portas de saída digitais do Arduino e foram utilizados para acionar as contadoras que acionam os circuitos de potência do sistema de resfriamento evaporativo do ar e do sistema de refrigeração convencional. O funcionamento das contadoras é similar ao funcionamento dos relés, porém as contadoras acionam correntes maiores do que as correntes que os relés podem acionar.

Foi utilizado um módulo relé 5V com 4 relés com as seguintes especificações:

- Modelo: SRD-05VDC0SL-C
- Tensão de operação: 5Vcc
- Permite controlar cargas de 220Vac
- Corrente típica de operação: 15~20mA
- Pinagem: Normalmente Aberto (NA), Normalmente Fechado (NF) e Comum (C)
- Tensão de saída: 30 Vcc a 10A ou 250Vac a 10A
- Tempo de resposta: 5~10ms



Figura 22. Módulo Relé 5V com 4 relés usado no projeto

Esse módulo foi alimentado externamente por um fonte de 5V e 1A para fornecer a corrente necessária para o seu acionamento e isolamento com a fonte que alimenta o Arduino.

3.4.5 Roteador

O roteador é um elemento intermediário em uma rede de computadores, que permite o roteamento de pacotes entre redes separadas. Este roteamento é realizado de acordo com um conjunto de regras que formam a tabela de roteamento. É um equipamento de camada 3 em relação ao modelo OSI [22].

Como a rede instalada nesse projeto é uma rede dedicada ao sistema de automação do sistema de climatização híbrida, pode-se afirmar que o roteador não está fazendo o roteamento entre redes separadas. Nesse caso, o dispositivo está funcionando como um Switch (ou comutador), que é uma ponte multiportas, o que quer dizer que se trata de um elemento ativo que age no nível 2 do modelo OSI [22]. Um Switch é um equipamento que interliga os computadores em uma rede, uma vez que os cabos de rede de cada computador se ligam a ele, que então direciona os dados enviados de um computador especificamente para outro.

As únicas partes do firmware específico do roteador que estão sendo utilizadas é sua tabela DHCP e a interface de rede sem fio. A tabela DHCP é usada apenas para distribuir IPs aos dispositivos móveis e computadores, para que esses acessem a interface gráfica do openHAB ou para fazer a configuração do sistema.

O roteador possui endereço de IP fixo 192.168.1.1



Figura 23. Roteador sem fio de 4 portas ethernet (RJ45)

3.5 IMPLEMENTAÇÃO DO PROTOCOLO DE COMUNICAÇÃO

Uma vez que o Mosquitto Broker encontra-se em execução no Raspberry Pi e todos os dispositivos do sistema estão na mesma rede, o openHAB e o Arduino devem estabelecer conexão com o *broker* para dar início à troca de mensagens.

```
pi@raspberrypi:~ $  
pi@raspberrypi:~ $ sudo service --status-all|grep -E "openh|mosquitto|mysql"  
[ + ] mosquitto  
[ + ] mysql  
[ + ] openhab  
pi@raspberrypi:~ $
```

Figura 24. Verificação dos estados serviços openHAB, Mosquitto e MySQL. O símbolo “[+]” indica que o serviço está sendo executado

Dessa forma, os sensores são lidos pelo Arduino, que publica seus valores em 7 tópicos diferentes:

- reuniao/temperatura - (TBS da sala de reuniões)
- reuniao/umidade - (umidade relativa da sala de reuniões)
- reuniao/TBU - (temperatura de bulbo úmido sala de reuniões)
- reuniao/tempdew - (temperatura de ponto de orvalho da sala de reuniões)
- externa/temperatura - (TBS do ar externo)
- externa/umidade – (umidade relativa do ar externo)
- externa/TBU - (temperatura de bulbo úmido do ar externo)

Os comandos para ligar e desligar os sistemas *split*, chamado de “condensador”, e o sistema evaporativo, chamado de “bomba”, são publicados em dois tópicos diferentes:

- condensador
- bomba

O openHAB está subscrito nos tópicos publicados pelo Arduino. Uma vez que o openHAB recebe as mensagens, elas são tratadas por conversão de tipos (*typecasting*) e guardadas em seus Itens para serem armazenadas no banco de dados e utilizadas pelas suas Regras.

A conversão de tipos é necessária uma vez que as mensagens em MQTT são transmitidas em ASCII. Dessa forma, o Arduino recebe a leitura dos sensores em ponto flutuante (*float*), converte-os para ASCII (vetor de char) e então publica cada valor em um tópico. Quando o openHAB recebe esses dados, eles são novamente convertidos de ASCII para ponto flutuante.

As formas como os dados são tratados pelo openHAB e pelo Arduino estão descritas nos Anexos 1 e 2 , respectivamente.

A Figura 25 ilustra um diagrama com o fluxo de mensagens trocadas pelo sistema de automação implementado.

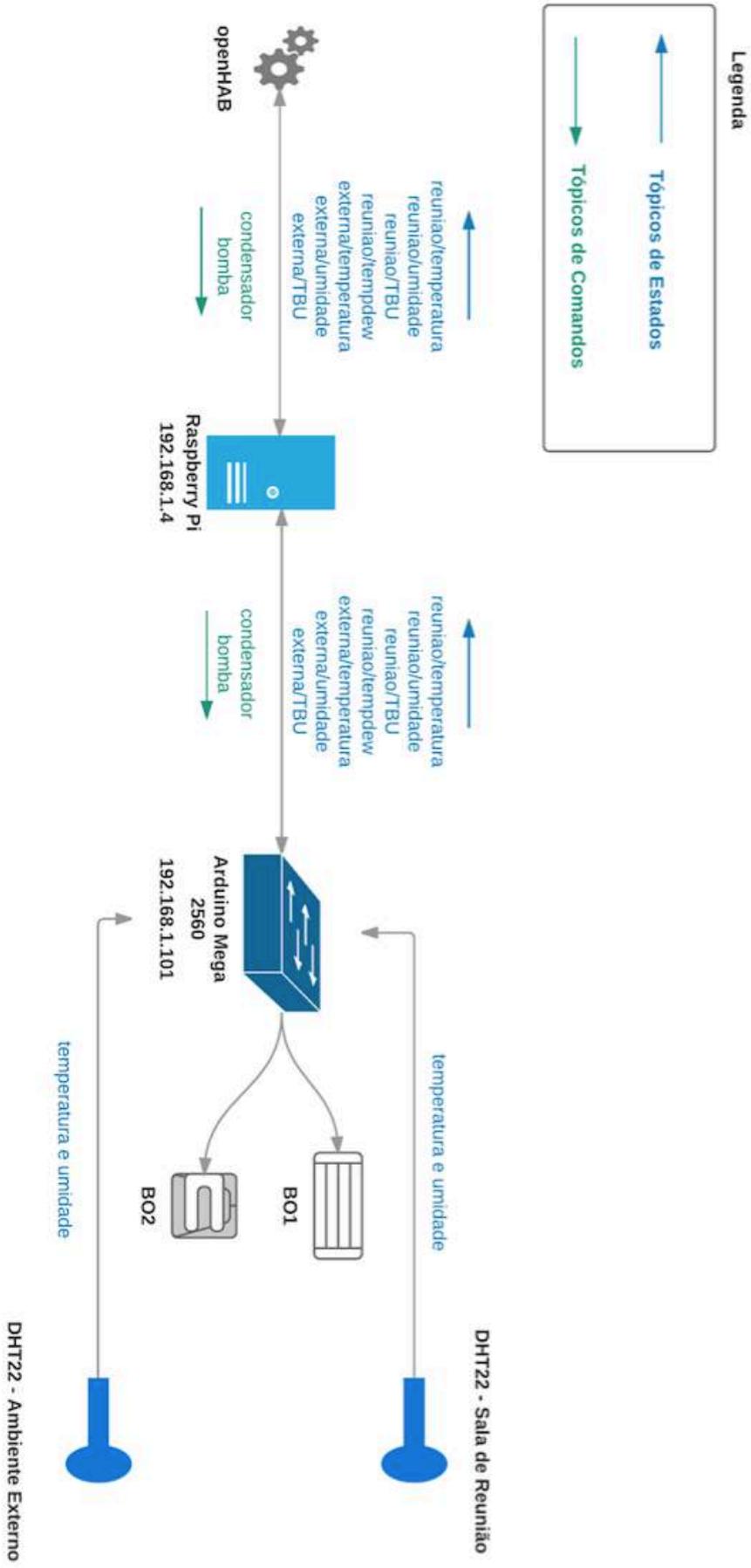


Figura 25. Diagrama de comunicação com o protocolo MQTT

3.6 PARÂMETROS DE PROJETO

3.6.1 CÁLCULO DE CONSUMO POR UNIDADE

O sistema de Climatização Híbrida da sala de reuniões do LARA pode ser dividido em quatro equipamentos ou grupo de equipamentos distintos:

1. Unidade Condensadora
2. Ventilador da Unidade Evaporadora
3. Bomba d'Água
4. Sistema de Automação (Raspberry Pi, Arduino, Módulo Relé, Roteador sem fio)

Conhecendo-se a carga nominal de cada um desses equipamentos e usando-se a Equação 11, a Tabela 9 pode ser elaborada.

Tabela 9. Consumo por equipamento

Item	Equipamento	Carga (W)	Consumo por minuto(KWh/min)
1	Unidade Condensadora e Atuador do Damper	2180	0,03633
2	Ventilador	127	0,00212
3	Bomba d'Água	8	0,000133
4	Sistema de Automação	10	0,00016

3.7 DESCRIÇÃO DE FUNCIONAMENTO

Conforme o diagrama blocos unifilar da Figura 14, o sistema de ar condicionado híbrido da sala de reuniões do LARA funcionará da seguinte forma:

O ventilador estará sempre ligado quando o disjuntor geral do sistema D2 estiver ligado. Essa escolha foi feita para promover a renovação constante do ar da sala.

Toda vez que o sistema de refrigeração *split* estiver ligado, o atuador do *damper* será acionado para a posição totalmente aberto. Quando esse sistema estiver desligado, o atuador do *damper* será acionado para a posição totalmente fechado. Quando o atuador do *damper* está na posição totalmente aberta, o ar da sala é forçado a recircular por dentro da unidade evaporadora e, por diferença de pressão (devido à manta do sistema evaporativo), o ar externo que entra na sala é mínimo. Essa situação aumenta a eficiência do refrigerador *split*. Quando o atuador do *damper* está na posição totalmente fechada, o ar é forçado pela manta do sistema evaporativo e acontece a renovação do ar da sala. Essa situação é importante por tratar-se de uma sala de com área 15m² com lotação máxima de 10 pessoas.

Segundo [23], a temperatura de bulbo úmido do ar em um sistema evaporativo nunca é atingida. Haverá sempre um **approach** na ordem de 5°C, em média. Dessa forma, para temperatura de conforto de 23°C, o sistema de resfriamento evaporativo só será ligado quando a temperatura de bulbo úmido do ar externo for em torno de

18°C (23°C - 5°C). Assim, se a temperatura de bulbo úmido do ar externo for de até 18°C, o uso sistema de refrigeração evaporativo será priorizado para que a temperatura de conforto de 23°C seja obtida, minimizando-se a energia elétrica gasta. Na prática, como será mostrado no Capítulo 4 desse trabalho, a maior diferença entre a temperatura interna da sala e a temperatura de bulbo seco do ar externo foi de 4,5°C.

3.7.1 IMPLEMENTAÇÃO DA LÓGICA DE AUTOMAÇÃO COM BASE NA TEMPERATURA DE BULBO ÚMIDO EXTERNA

O sistema de automação controlará a temperatura da sala em torno de 23°C com a implementação dos *set-points* de 23,5°C e 23°C. A escolha desse set-point inferior de 23°C é devido ao fato de que a temperatura sempre ultrapassa esse limiar quando o ar está sendo resfriado. Se a temperatura de bulbo úmido externa for menor do que 19,5°C, o sistema ligará a bomba d'água para causar a diminuição da temperatura. Se a TBU externa for maior do que 19,5°C, o compressor do sistema *split* será ligado.

Conforme a norma ISO 9424, a umidade relativa do ar mínima para conforto é de 40%. Dessa forma, o sistema de refrigeração evaporativo será ligado mesmo com temperaturas abaixo dos 22,5°C para que esse mínimo de umidade relativa do ar seja atendida.

Como forma de segurança, um limiar máximo de 24,2°C foi definido para que o sistema de resfriamento convencional seja ligado caso a lógica implementada não consiga controlar a temperatura da sala.

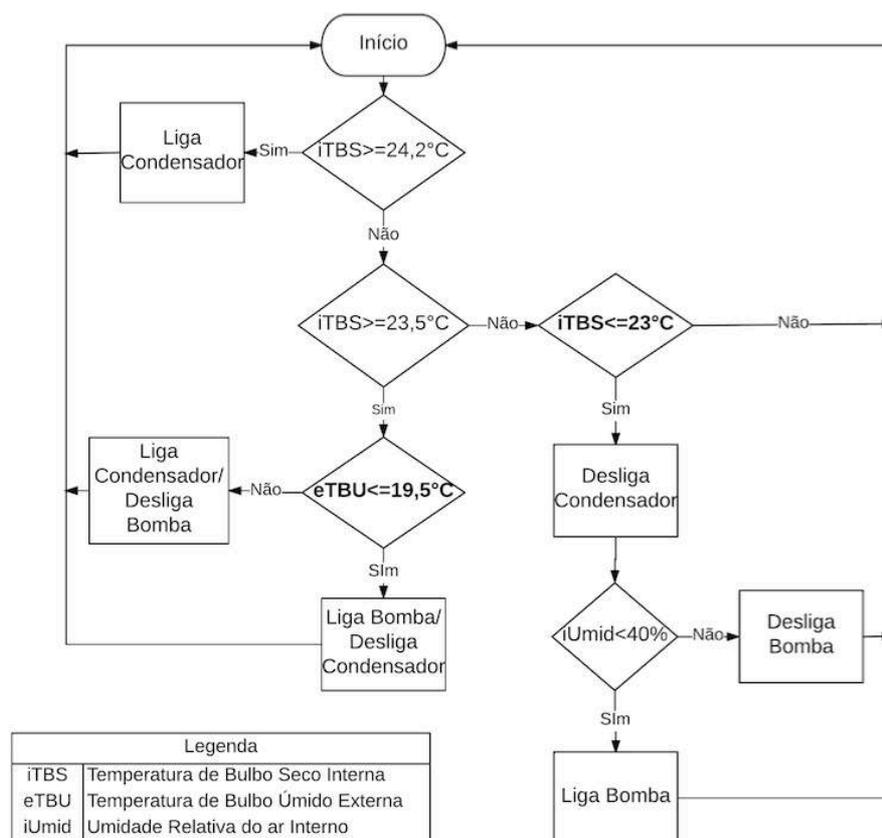


Figura 26. Diagrama de Fluxo da Lógica de Automação do Sistema de Ar Condicionado Híbrido da Sala de Reuniões do LARA

CAPÍTULO 4 – RESULTADOS E ANÁLISE

Nesse capítulo serão apresentados os resultados práticos obtidos com o sistema de automação. Serão exibidos os dados colhidos com os sensores do sistema, além dos resultados de consumo e atuação do sistema de automação. Todos os dados foram gravados pelo openHAB em um banco de dados MySQL e são mostrados através da interface gráfica para openHAB chamada HABmin.

4.1 TEMPERATURA DE BULBO ÚMIDO EXTERNA

A temperatura de bulbo úmido do ar externo é um fator crucial para o desempenho e rendimento do sistema de refrigeração por umidificação. Conforme [23], o **approach** do resfriamento do ar usando-se esse sistema será em torno de 5°C. Dessa forma, para a temperatura de conforto de 23°C, a temperatura de bulbo úmido do ar externo deve ser de até 18°C. A Figura 27 mostra a progressão das temperaturas de bulbo seco e úmido do ar externo.

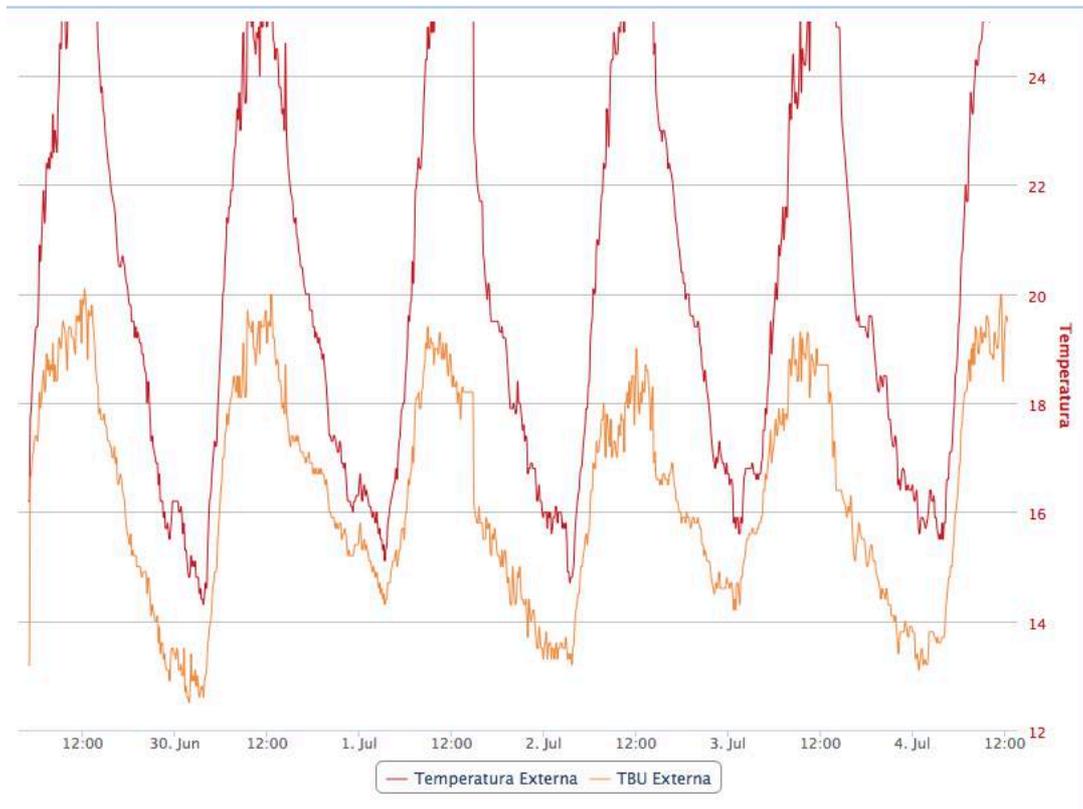


Figura 27. Comportamento das temperaturas externas de bulbo seco (Temperatura Externa – em vermelho) e de bulbo úmido (TBU Externa – em Amarelo) durante 5 dias

Conforme a Figura 27, percebe-se que a temperatura de bulbo úmido externa fica acima dos 18°C durante uma pequena faixa de horários durante os dias. Essa faixa de horário costuma ser entre 7h e 14h, nos dias de **inverso**. Durante o verão, essa faixa de horário aumenta. Dessa forma, o maior potencial do uso do sistema de refrigeração evaporativo dar-se-á nos demais horários. As Figuras 28 e 29 mostram a variação da temperatura de bulbo úmido durante os dias de forma mais detalhada.

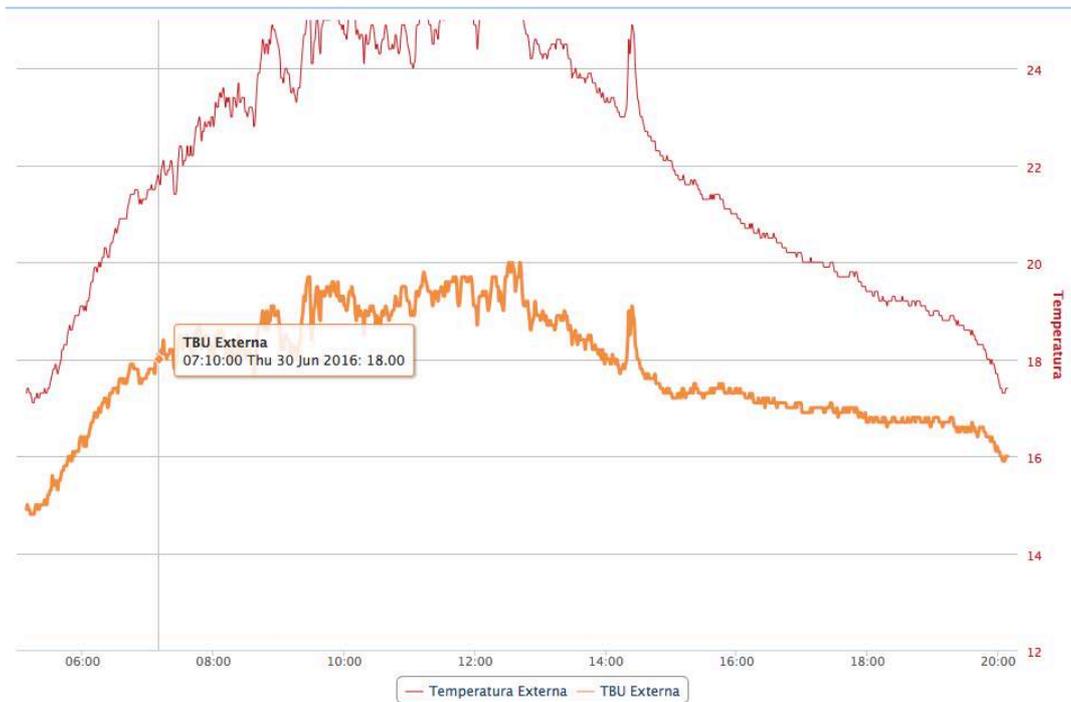


Figura 28. Comportamento das temperaturas externas de bulbo seco (Temperatura Externa – em vermelho) e de bulbo úmido (TBU Externa – em Amarelo) durante um dia. A TBU Externa ultrapassa a barreira dos 18°C às 7h10. Essa temperatura volta a ficar abaixo dos 18°C em torno de 14h.

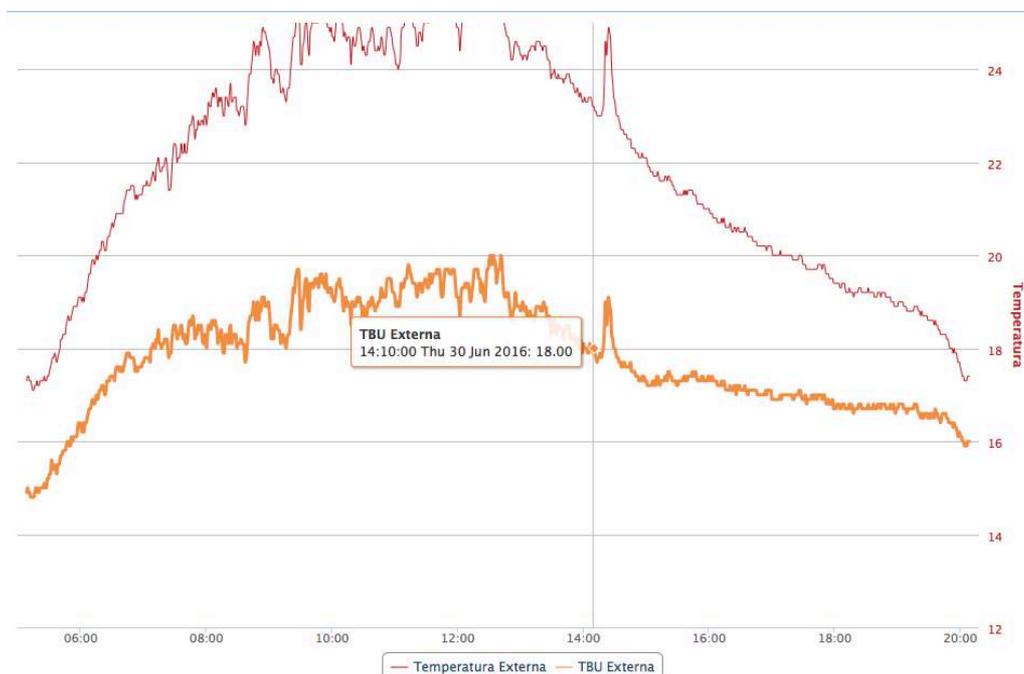


Figura 29. Comportamento das temperaturas externas de bulbo seco (Temperatura Externa – em vermelho) e de bulbo úmido (TBU Externa – em Amarelo) durante um dia. A TBU Externa ultrapassa a barreira dos 18°C em torno das 7h. Essa temperatura volta a ficar abaixo dos 18°C em torno de 14h10.

Em geral, a Figura 30 mostra os comportamentos da umidade externa e das temperaturas de bulbo úmido e bulbo seco.

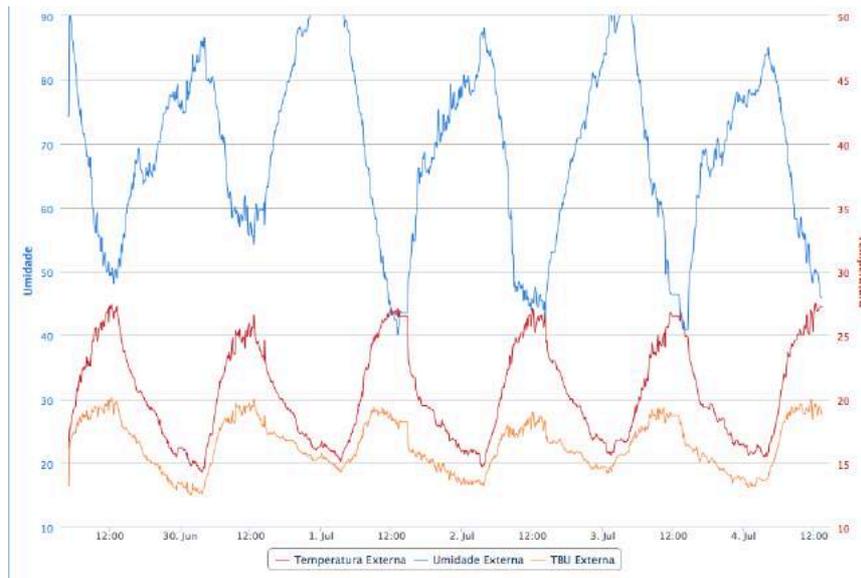


Figura 30. Comportamento da umidade externa e das temperaturas externas de bulbo seco (Temperatura Externa – em vermelho) e de bulbo úmido (TBU Externa – em Amarelo) durante 5 dias.

Usando-se o sistema de resfriamento evaporativo e mantendo-se o sistema de refrigeração *split* desligado, ou seja, ligando-se a bomba d'água e desligando-se o compressor do sistema *split*, observa-se que o *approach* da temperatura de bulbo seco do ar interno, na prática, é menor do que 5°C. Este fenômeno deve-se ao fato de o *approach* [23] ser um limite médio. A Figura 31 demonstra esse fato.

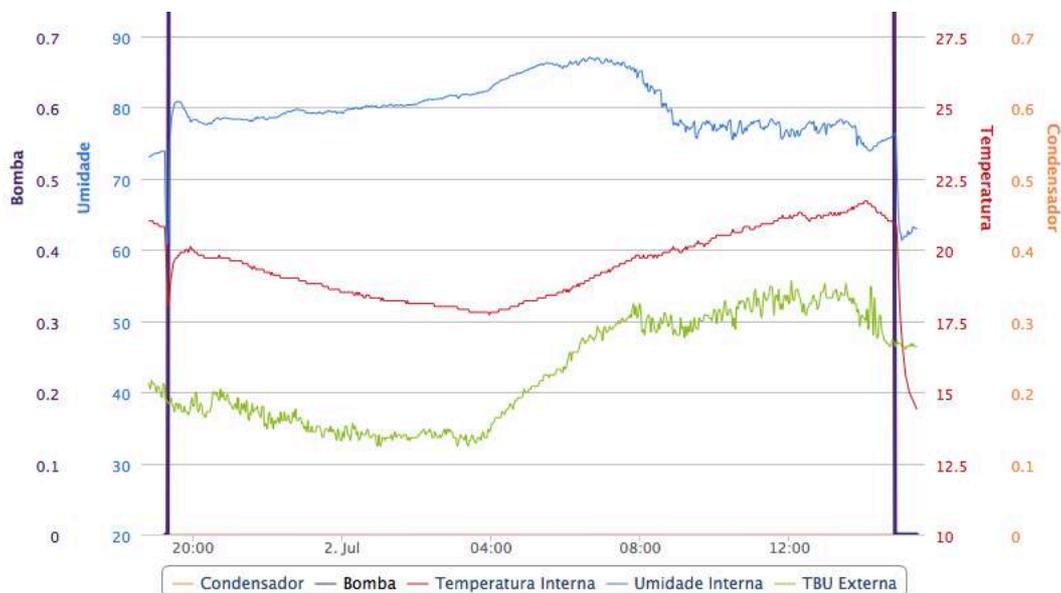


Figura 31. Comportamento da variação da temperatura de bulbo seco interna (Temperatura Interna – em vermelho) quando se mantém somente a bomba d'água ligada. Nesse intervalo, a maior diferença entre a temperatura interna e a temperatura de bulbo úmido externa foi de 4,5°C.

4.1. LÓGICA DE AUTOMAÇÃO IMPLEMENTADA POR OLMOS

Em sua ~~tese~~ de mestrado, Olmos [1] implementou a lógica descrita na Tabela 10 para controlar o sistema de climatização híbrida da sala de reuniões do LARA.

Tabela 10. Parâmetros de controle do sistema de climatização híbrida [1]

Modo de Operação: Híbrido		
Ação	Set-point 1	Set-point 2
	Compressor	Bomba d'água
Liga	23,5°C	60%
Desliga	22,5°C	65%

A Figura 32 mostra essa lógica de automação implementada.

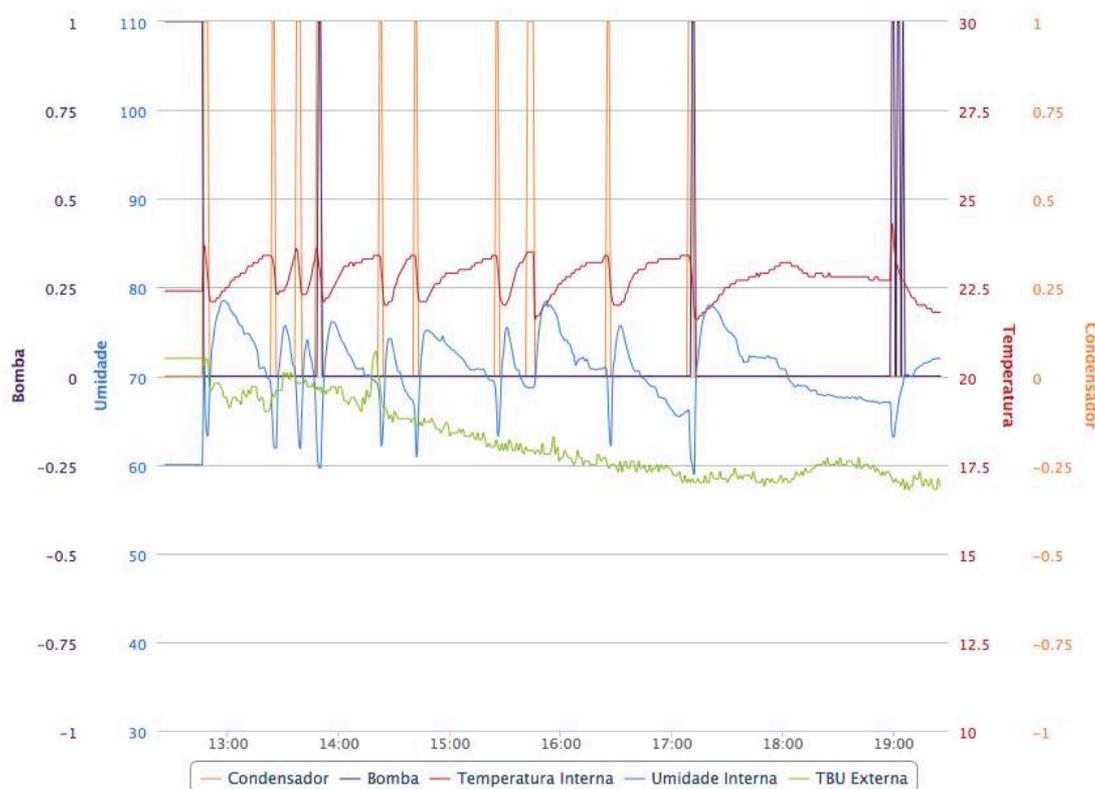


Figura 32. Lógica de controle implementada por Olmos em sua tese de mestrado.

Esta lógica leva em consideração apenas a temperatura e a umidade relativa do ar da sala de reuniões. Dessa forma, não é possível prever na queda de temperatura interna da sala com o uso do sistema evaporativo.

Mesmo com essas condições, Olmos [1], no ano de 2009, conseguiu com o sistema de climatização híbrida a redução de 67,1% no consumo de energia, se comparado com o sistema *split* regulando a temperatura em torno dos 23°C.

Analisando-se a Figura 33, percebe-se que mesmo com grande potencial de uso do sistema evaporativo (TBU Externa < 18°C), a lógica ignora esse fato e continua a usar o sistema convencional, que consome mais energia.

A diminuição do consumo de energia dessa lógica advém do fato de a umidade relativa do ar externo no período ter ficado entre 75% e 25% e a temperatura de externa, entre 28°C e 16°C entre os dias 23/06/2009 e 24/06/2009, Figura 33.

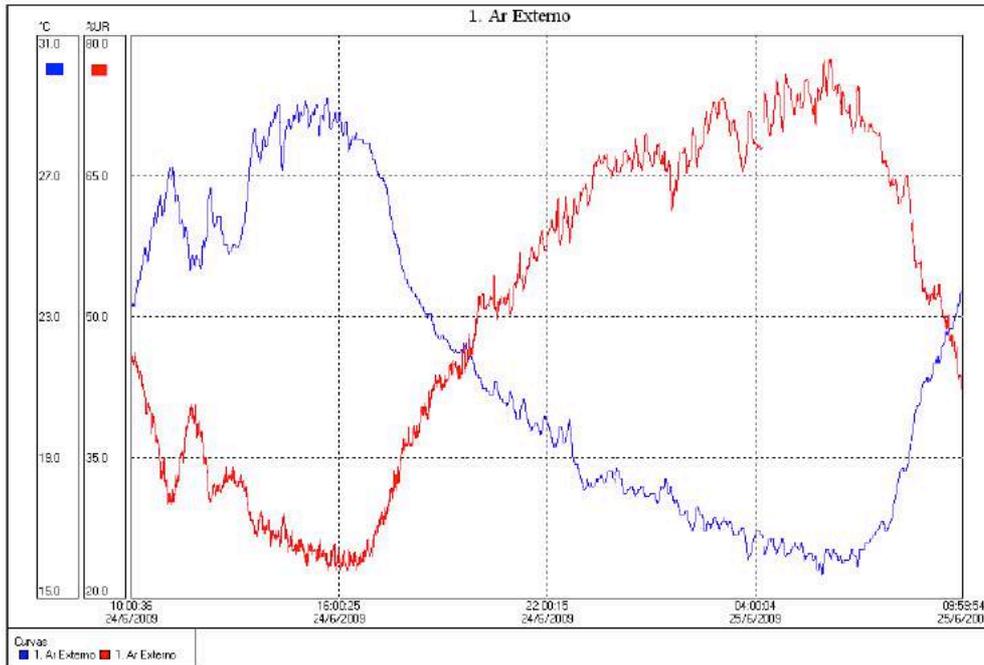


Figura 33. Gráfico da Temperatura e Umidade do ar Externo entre os dias 23/06/2009 e 24/06/2009

Entre os dias 05/07/2016 e 06/07/2016, a implementação dessa lógica levou ao consumo de 3,16 kWh, uma vez que a umidade relativa do ar externo variou entre 94,4% e 42,3% e a temperatura externa variou entre 28,9°C e 15,6°C, Figura 34.

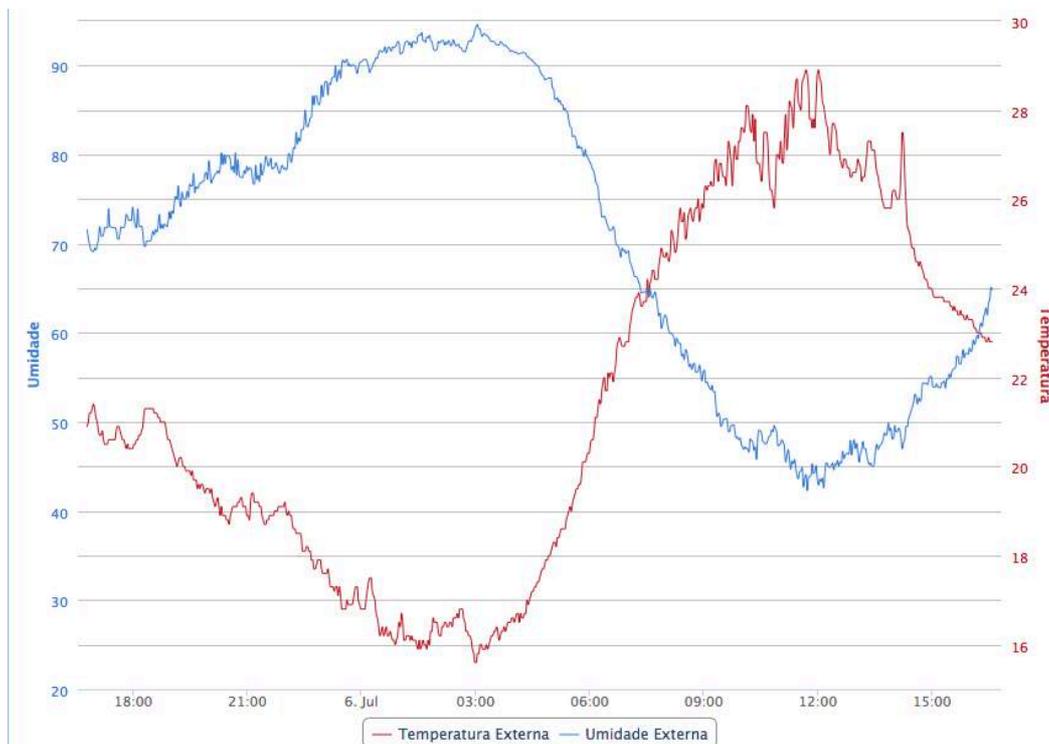


Figura 34. Gráfico da Temperatura e Umidade do ar Externo entre os dias 05/07/2016 e 06/07/2016.

TG Daniel Bello	
Status	
Arduino	online
Condensador	Ligado
Bomba	Desligado
Sensores	
Temperatura Sala	23.4 °C
Umidade Sala	66.1 %
Temp. Orvalho	16.7 °C
Temperatura Externa	20.7 °C
Umidade Externa	71.3 %
TBU Externa	17.1 °C
©2010-2015 openHAB.org	

Figura 35. Interface gráfica do openHAB mostrando o sistema de refrigeração convencional ligado, mesmo com a TBU do ar externo em 17,1°C

O consumo aferido com o medidor de energia *Landis + Gyr*, Figura 36, apenas definindo-se set-points para o sistema de refrigeração *split*, foi de 7,52KWh. No ano de 2009, a mesma medição foi de 5,92KWh. Essa diferença deve-se tanto às diferenças climáticas entre os anos, quanto à falta de manutenção dos equipamentos e sua deterioração natural.



Figura 36. Detalhes do medidor de energia *Landis + Gyr*

Com essa nova realidade de consumo, essa lógica de automação consumiu 3,16 KWh, ou seja, uma redução de 57,98% no consumo.

Tabela 11. Comparação de Consumo do sistema no ano de 2009 e 2016

Ano	Consumo da Lógica Sistema Split (KWh)	Consumo da Lógica implementada por Olmos (KWh)	Redução de consumo
2009	5,92	1,95	67,07%
2016	7,52	3,16	57,98%

4.2 LÓGICA DE AUTOMAÇÃO COM BASE NA TEMPERATURA DE BULBO ÚMIDO DO AR EXTERNO

Implementando-se a lógica de controle que leva em consideração a temperatura de bulbo úmido do ar externo, Figura 26, percebe-se que o sistema de climatização híbrido prioriza muito mais o resfriamento por evaporação, Figura 37.

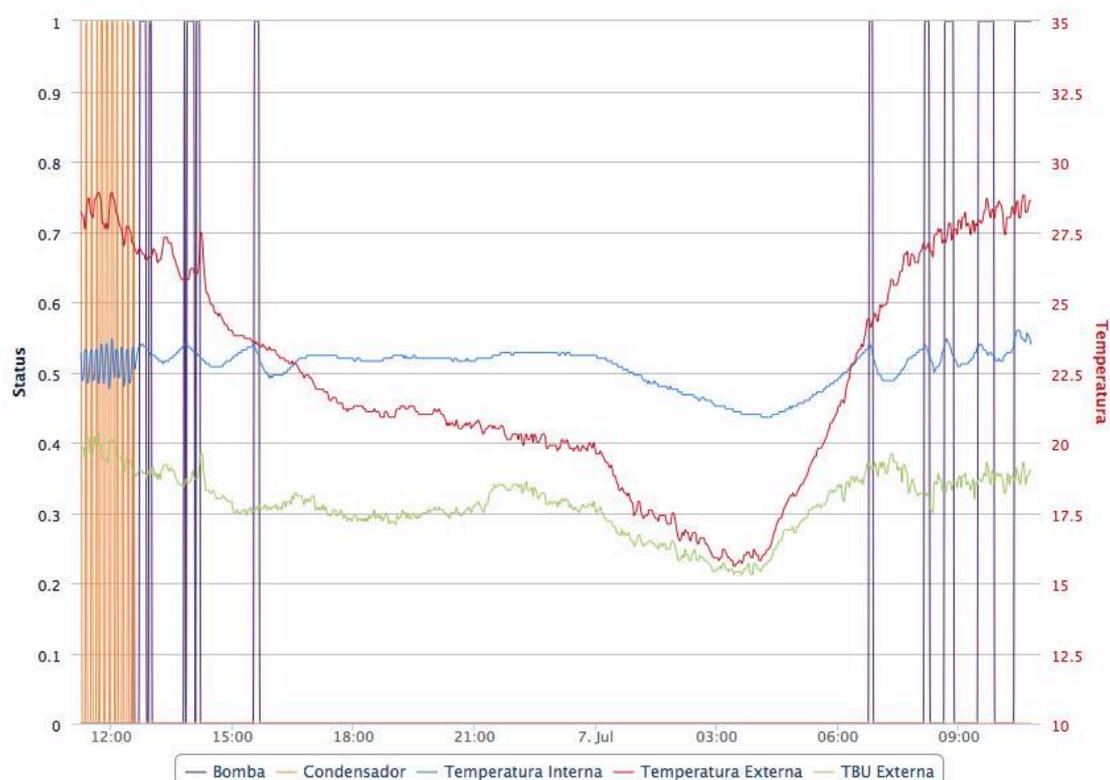


Figura 37. Implementação da lógica de automação com base na TBU do ar externo

As linhas de cor laranja e roxa, representam, respectivamente, o estados ligado ou desligado dos sistemas evaporativo e convencional, com as legendas “Bomba” e “Condensador”.

Percebe-se na Figura 37 que, quando a temperatura de bulbo úmido do ar externo encontra-se acima dos 19,5°C, a preferência foi dada ao sistema convencional de refrigeração. Porém, quando essa temperatura diminuiu ao longo do dia, o controle da temperatura do ar da sala de reuniões pode ser realizado com sucesso somente com o sistema evaporativo.

The image shows a screenshot of the openHAB web interface for a system named 'TG Daniel Bello'. The interface is divided into two main sections: 'Status' and 'Sensores'. The 'Status' section contains three rows: 'Arduino' (online), 'Condensador' (Desligado), and 'Bomba' (Ligado). The 'Sensores' section contains six rows: 'Temperatura Sala' (23.4 °C), 'Umidade Sala' (64.0 %), 'Temp. Orvalho' (16.2 °C), 'Temperatura Externa' (29.3 °C), 'Umidade Externa' (37.5 %), and 'TBU Externa' (19.4 °C). At the bottom, there is a copyright notice: '©2010-2015 openHAB.org'.

TG Daniel Bello	
Status	
Arduino	online
Condensador	Desligado
Bomba	Ligado
Sensores	
Temperatura Sala	23.4 °C
Umidade Sala	64.0 %
Temp. Orvalho	16.2 °C
Temperatura Externa	29.3 °C
Umidade Externa	37.5 %
TBU Externa	19.4 °C
©2010-2015 openHAB.org	

Figura 38. Interface gráfica do openHAB mostrando o sistema de refrigeração evaporativo ligado e controlando a temperatura interna

O consumo aferido com o medidor de energia Landis + Gyr para essa lógica de automação foi de 2,41 KWh, alcançando uma redução de 73,32%.



Figura 39. Interface para telefone celular do openHAB mostrando situação em que a TBU Externa está maior do que 19,5 e a temperatura interna está maior que 23,6. Nesse caso, o sistema convencional é ligado e o evaporativo, desligado.

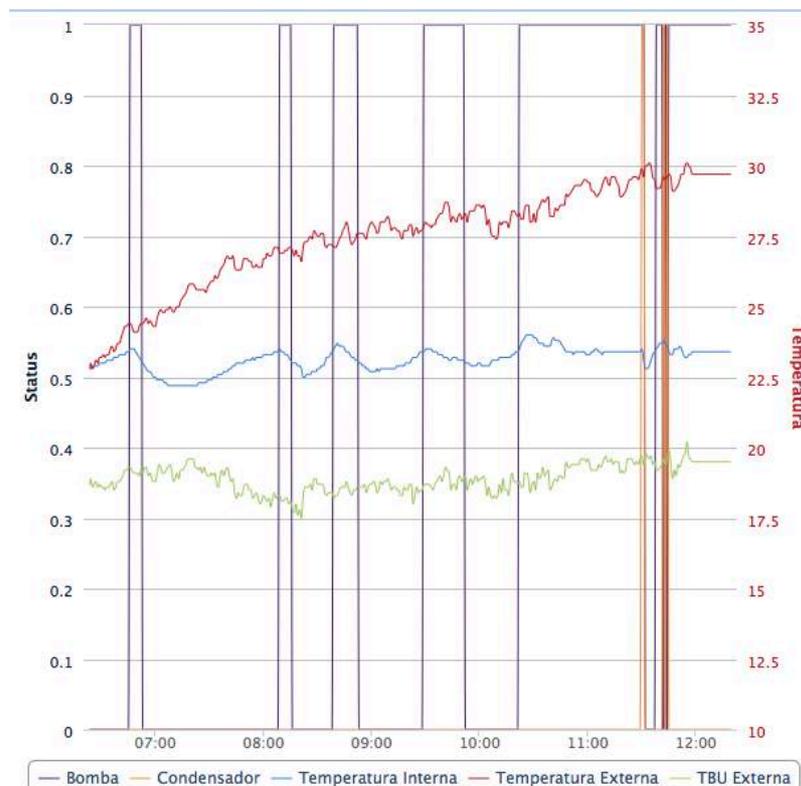


Figura 40. Situação em que a TBU Externa aumentou e o sistema convencional teve que ser ligado.

CAPÍTULO 5 – CONCLUSÕES

5.1. CONCLUSÕES GERAIS

Nearly Zero Energy Buildings (NZEB) ou Edifícios com consumo de energia próximo de zero é a nova aposta para o conceito de cidade inteligente. O conceito diz respeito à um edifício com um desempenho energético muito elevado em que as necessidades de energia quase nulas ou muito pequenas deverão ser cobertas por energias renováveis. A ideia é de extrema importância para o atual contexto mundial e brasileiro, visto que o consumo está sempre próximo da capacidade produtiva, o que torna a prática de racionamento cada vez mais comum para evitar-se o colapso do sistema energético.

Conseguir um saldo quase nulo de energia usando-se grandes sistemas de energias renováveis, para compensar um nível de consumo elevado, seria uma aberração. O método correto para iniciar-se um projeto de NZEB, começa por exigir uma redução de consumo a um valor ~~máximo~~ aceitável para depois passar a abastecê-lo com energias renováveis.

Segundo [24], os edifícios são os responsáveis por 40% do consumo de energia elétrica do país. Do consumo energético dos edifícios, 48% são gastos com sistemas de ar condicionado [25]. Assim, o consumo brasileiro com sistemas de ar condicionado em edifícios representa 19,2% do total produzido no país.

Sob essas circunstâncias, o sistema de climatização híbrida instalado na sala de reuniões do LARA mostra-se como uma alternativa viável para alcançar-se redução no consumo de energia desses edifícios.

A redução alcançada de 73,32% levaria a uma redução de 5,12% do consumo energético do Brasil. Como comparação, a meta de redução da implantação do horário de verão no Brasil é de 0,9%.

Além da redução do consumo energético, o sistema híbrido de climatização promove a constante renovação do ar da sala de reuniões, além de garantir os valores para conforto térmico exigidos pelas Normas Técnicas.

5.1. PERSPECTIVAS PARA PESQUISAS FUTURAS

Como forma de promover a melhoria do sistema de climatização híbrida, as seguintes medidas são sugeridas:

- O uso de um sistema de refrigeração *split* dutável do tipo *inverter*, ou seja, usar um sistema que permita o funcionamento na velocidade adequada para garantir a temperatura desejada.
- Desenvolver um controle proporcional do atuador do *dampers*, para promover o controle da mistura de ar interno e externo da sala, para que as condições de temperatura e umidade resultantes permitam tirar maior proveito dos dois sistemas instalados
- Aumentar a área de contato entre o ar de entrada e a manta molhada, para que a quantidade de umidade injetada no ar seja aumentada. Essa atitude permitirá diminuir a temperatura de *approach* entre a temperatura interna e a temperatura de bulbo úmido do ar exterior da sala.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] OLMOS, J. L. Sistema Híbrido de Climatização Visando Conforto Térmico e Eficiência Energética. Dissertação de Mestrado em Engenharia Elétrica, Publicação PPGEENE. DM393/09, Departamento de Engenharia Elétrica, Universidade de Brasília, DF.
- [2] Bhatia, B.E. Principles of Evaporative Cooling System
- [3] ASHRAE. Handbook of Fundamentals. Atlanta-GA, American Society of Heating, Refrigerating and Air-Conditioning Engineers, 2005.
- [4] ASHRAE. Handbook of systems and equipment. Atlanta-GA, American Society of Heating, Refrigerating and Air-Conditioning Engineers, 2000
- [5] GOSNEY, W. B. Principles of refrigeration. London, Cambridge University Press, 1982.
- [6] STOECKER, W. F. ; JONES, J. W. Refrigeração e ar condicionado. São Paulo, McGraw-Hill, 1985.
- [7] STULL, R. Wet-Bulb Temperature from Relative Humidity and Air Temperature, Vancouver, British Columbia, Canada, 2011.
- [8] BOLTON, D. Atmospheric Physics Group, Imperial College, London, England, 1980.
- [9] POTTER, C. M. SCOTT, P. E. Ciências Térmicas: Termodinâmica, Mecânica dos Fluidos e Transmissão de Calor. 1a Edição. São Paulo: Thomson Learning, 2007.
- [10] MILLER, R. MILLER, R. M. Refrigeração e Ar Condicionado. 1a Edição. Rio de Janeiro. LTC, 2008.
- [11] CURSO DE REFRIGERAÇÃO E AR CONDICIONADO. Parte 1. Rio Grande do Sul:Edubras, 2008.
- [12] LAMPKIN, V. Building Smarter Planet Solutions with MQTT and IBM WebSphere MQ Telemetry, North Castle Drive, Armonk, NY, USA. 2012
- [13] GITHUB. openHAB Documentation. Disponível em <https://github.com/openhab/openhab/wiki#overview> > Acesso em: 6 de junho 2016
- [14] STENMAN, E. The Erlang Run-Time System, O'Reilly Media, 2014.
- [15] DUBOIS, P. MySQL Developer's Library. Fifth Edition. Upper Saddle River, NY, USA. 2013.
- [16] MUNTERS, Painéis De Resfriamento Evaporativo Celdek, Munters Brasil Indústria e Comércio Ltda. Curitiba.2008.
- [17] SPRINGER C., Manual de Instalação, Operação E Manutenção do Split Versatile, Springer Carrier Ltda. Canoas – RS.2008.
- [18] CIBRACON, Manual do Sensor Controlador de Temperatura. 2008
- [19] ISO 9424, Wood-based panels, Determination of dimensions of test
- [20] NBR 17, Ergonomia. 1978.
- [21] CAVALIN, G. Instalações Elétricas Prediais. 12 Edição. 2014

- [22] ODOM, W. Networking Basics. Cisco Press, 2015.
- [23] CREDER, H. Instalações de Ar Condicionado. 6 Edição, 2004
- [24] EPE. Balanço Energético Nacional, ano base 2014. 2015.
- [25] PROCEL. Manual de Conservação de Energia Elétrica em Prédios Comerciais e Públicos. Rio de Janeiro. 2005

ANEXOS

Anexo 1 – Códigos de Configuração do openHAB

Anexo 2 – Código de Configuração do Arduino

1. Items – Arquivo javaone.items

```

Switch Condensador "Condensador" <climate> {
mqtt=">[mosquitto:condensador:command:*:default]" }
Number StatusCondensador
"Condensador[MAP(ligadoDesligado.map):%s]"
Switch Bomba "Bomba" <bath> {
mqtt=">[mosquitto:bomba:command:*:default]" }
Number StatusBomba "Bomba[MAP(ligadoDesligado.map):%s]"

Number TemperaturaInterna "Temperatura Interna [%].1f
°C]" <temperature> {
mqtt="<[mosquitto:reuniao/temperatura:state:default]" }
Number UmidadeInterna "Umidade Interna [%].1f %" <water>
{ mqtt="<[mosquitto:reuniao/umidade:state:default]" }
Number TBUInterna "TBU Interna [%].1f °C]" {
mqtt="<[mosquitto:reuniao/tbu:state:default]" }
Number TempOrvalho "Tempertura Orvalho [%].1f °C]" {
mqtt="<[mosquitto:reuniao/tempdew:state:default]" }

Number TemperaturaExterna "Temperatura Externa [%].1f
°C]" <temperature> {
mqtt="<[mosquitto:externa/temperatura:state:default]" }
Number UmidadeExterna "Umidade Externa [%].1f %" <water>
{ mqtt="<[mosquitto:externa/umidade:state:default]" }
Number TBUExterna "TBU Externa [%].1f °C]" {
mqtt="<[mosquitto:externa/tbu:state:default]" }
Number SetpointT "SP(Temp.) [%].1f °C]" <climate>

Switch StatusArduino "Arduino [MAP(conexao.map):%s]" (Status,
Network) { nh="192.168.1.101" }

```

2. Sitemap – Arquivo default.sitemap

```
sitemap default label="TG Daniel Bello" {  
  
    Frame label= "Status" {  
  
        Text item=StatusArduino  
        Text item=StatusCondensador  
        Text item=StatusBomba  
    }  
  
    Frame label= "Sensores" {  
        Text item=TemperaturaInterna label="Temperatura  
Sala [%].1f °C]"  
        Text item=UmidadeInterna label="Umidade Sala  
[%].1f %]"  
  
        Text item=TempOrvalho label="Temp. Orvalho [%].1f  
°C]"  
  
        Text item=TemperaturaExterna label="Temperatura  
Externa [%].1f °C]"  
        Text item=UmidadeExterna label="Umidade Externa  
[%].1f %]"  
        Text item=TBUExterna label="TBU Externa [%].1f  
°C]"  
  
    }  
}
```

3. Rules – Arquivo javaone.rules

```
import org.openhab.core.library.types.*
import org.openhab.core.library.types.DecimalType

rule "Iniciar valores virtuais dos itens"
when
    System started
then
    if (SetpointT.state == Uninitialized) {
        SetpointT.postUpdate(25)
    }
end

rule "Status Bomba"
when
    Item Bomba received command or
    Time cron "0 0/1 * * * ?"
then
    if(receivedCommand==ON) {
        postUpdate(StatusBomba,1)
    }
    else {
        postUpdate(StatusBomba,0)
    }
    if(Bomba.state==ON){
        postUpdate(StatusBomba,1)
    }
    else{
        postUpdate(StatusBomba,0)
    }
end

rule "Status Condensador"
when
    Item Condensador received command or
    Time cron "0 0/1 * * * ?"
then
    if(receivedCommand==ON) {
        postUpdate(StatusCondensador,1)
    }
    else {
        postUpdate(StatusCondensador,0)
    }
}

if(Condensador.state==ON){
    postUpdate(StatusCondensador,1)
}
```

```

    }
    else{
        postUpdate(StatusCondensador,0)
    }
end

rule "Ar Condicionado "
when
    Item TemperaturaInterna changed or Item UmidadeInterna
changed
then
    var double temp = (TemperaturaInterna.state as
DecimalType).doubleValue
    var double umid = (UmidadeInterna.state as
DecimalType).doubleValue
    var double tbu = (TBUExterna.state as
DecimalType).doubleValue

    if(temp >= 23.5){
        if(tbu<=19.5){
            sendCommand(Bomba,ON)
            sendCommand(Condensador,OFF)
        }
        else{
            sendCommand(Condensador,ON)
            sendCommand(Bomba,OFF)
        }
    }
    if(temp <=23){
        if(umid<40){
            sendCommand(Bomba,ON)
        }
        else{
            sendCommand(Bomba,OFF)
        }
        sendCommand(Condensador,OFF)
    }
    if(temp>=25){
        sendCommand(Condensador,ON)
    }
}
end

```

4. Banco de Dados – Arquivo mysql.persist

```
// Persistence strategies have a name and a definition and are referred to in the "Items" section
```

```
Strategies {  
    everyMinute      : "0 * * * * ?"  
    everyHour       : "0 0 * * * ?"  
    everyDay        : "0 0 0 * * ?"
```

```
    // If no strategy is specified for an item entry below,  
    the default list will be used.
```

```
    default = everyChange  
}
```

```
Items {  
    /* : strategy = everyMinute, restoreOnStartup  
    TemperaturaInterna : strategy = everyMinute  
    UmidadeInterna : strategy = everyMinute  
    TBUInterna : strategy = everyMinute  
  
    TemperaturaExterna : strategy = everyMinute  
    UmidadeExterna : strategy = everyMinute  
    TBUExterna : strategy = everyMinute  
  
    StatusCondensador : strategy = everyMinute  
    StatusBomba : strategy = everyMinute  
  
    SetpointT: strategy = everyDay  
}
```

```
#include <DHT.h>
#include <math.h>
#include <PubSubClient.h>
#include <UIPEthernet.h>

// DHT Interno-----
const int DHTPIN = 3; // pino digital onde o DTH está conectado
#define DHTTYPE DHT22 // Definição do tipo de sensor
DHT dht(DHTPIN, DHTTYPE); //Inicialização do sensor
// DHT Interno-----

// DHT Externo-----
const int DHTPIN2 = 4; // pino digital onde o DTH está conectado
#define DHTTYPE2 DHT22 // Definição do tipo de sensor
DHT dht2(DHTPIN2, DHTTYPE2); //Inicialização do sensor
// DHT Externo-----

//Tempo para atualizar os sensores-----
unsigned long temperature_time;
//TEMPO-----

// IPS-----
byte mac[] = { 0xDE, 0xED, 0xBA, 0xFE, 0xFE, 0xED };
byte server[] = { 192, 168, 1, 4 }; //IP do Broker MQTT
byte ip[] = { 192, 168, 1, 101 }; //IP desse arduino
//IPS-----

//PINAGEM-----
#define condensador 22
#define damper 24 //ventilador
#define bomba 26
//define damper 28 //umidificador
#define resetArduino 45
int countReconnect=0;
//PINAGEM-----
```

```

//MQTT-----
EthernetClient ethClient;
PubSubClient client(ethClient);
//MQTT-----

//COMANDOS-----
void comandoCondensador(int comando){
  if(comando == 1){
    digitalWrite(condensador,HIGH);
    client.publish("condensador","condensador Ligado");
    // Serial.println("Ventilador");
  }
  if(comando == 0){
    digitalWrite(condensador,LOW);
    client.publish("condensador","condensador Desligado");
  }
}

void comandoBomba(int comando){
  if(comando == 1){
    digitalWrite(bomba,HIGH);
    client.publish("bomba","bomba ligada");
  }
  if(comando == 0){
    digitalWrite(bomba,LOW);
    client.publish("bomba","bomba desligada");
  }
}

//COMANDOS-----

// Callback function-----
void callback(char* topic, byte* payload, unsigned int length) {
  byte* pl = (byte*)malloc(length);
  memcpy(pl,payload,length);
  char Comando1 = (char)payload[0];
  char Comando2 = (char)payload[1];
  char Comando3 = (char)payload[2];
}

```

```

if(length == 2){
  if ((Comando1 =='O')&&(Comando2=='N')) {
    if((strcmp(topic,"condensador")==0))
      comandoCondensador(1);
    if((strcmp(topic,"bomba")==0))
      comandoBomba(1);
  }
}
if(length == 3){
  if ((Comando1 =='O')&&(Comando2=='F')&&(Comando3 =='F')) {

    if((strcmp(topic,"condensador")==0))
      comandoCondensador(0);
    if((strcmp(topic,"bomba")==0))
      comandoBomba(0);
  }
}

free(pl);
}
//CALLBACK-----

/--Função para reconectar Arduino ao Broker, caso a conexão caia
void reconnect() {
  // Loop until we're reconnected
  while (!client.connected()) {
    if(countReconnect==5){
      countReconnect==0;
      Serial.println("Resetei");
      //softReset();
    }
    else{
      countReconnect++;
    }
  }
  // Attempt to connect
  if (client.connect("arduinoClient")) {

```

```

    countReconnect=0;
    // Once connected, publish an announcement...
    client.publish("test","reconectado");
    // ... and resubscribe
    client.subscribe("test");
    client.subscribe("damper");
    client.subscribe("condensador");

    client.subscribe("bomba");
// Tem que ler último resultado do comando Geral na EEPROM
} else {
    // Wait 1 seconds before retrying
    delay(1000);
}
}
}
//RECONNECT-----

// SETUP-----
void setup()
{
    pinMode(condensador,OUTPUT);
    pinMode(bomba,OUTPUT);
    pinMode(resetArduino,OUTPUT);

    client.setServer(server, 1883);
    client.setCallback(callback);

    Ethernet.begin(mac, ip);
    if (client.connect("arduinoClient")) {
        client.publish("test","Arduino Conectado");
        client.subscribe("test");
        client.subscribe("damper");
        client.subscribe("condensador");
        client.subscribe("bomba");
    }
}

```

```

    dht.begin(); //dht interno
    dht2.begin(); //dht externo
    temperature_time = millis();

}
//SETUP-----

//LOOP-----

void loop(){

    unsigned long time_passed = 0;

    if (!client.connected()) {
        reconnect();
    }
    client.loop();

    time_passed = millis() - temperature_time;
    if (time_passed < 0)
    {
        temperature_time = millis();
    }

    if (time_passed > 5000)
    {

        //Temperatura e Umidade interna
        float h = dht.readHumidity();
        char umidadeInterna[10];
        dtostrf(h,4, 1, umidadeInterna);
        // Read temperature as Celsius
        float t = dht.readTemperature();
        char temperaturaInterna[10];
        dtostrf(t,4, 1, temperaturaInterna);

        //Temperatura e umidade externa

```

```

float h2 = dht2.readHumidity();
char umidadeExterna[10];
dtostrf(h2,4, 1, umidadeExterna);
// Read temperature as Celsius
float t2 = dht2.readTemperature();
char temperaturaExterna[10];
dtostrf(t2,4, 1, temperaturaExterna);

//Calculo TBU
float tbuOut = t2*atan(0.151977*pow((h2+8.313659),0.5))+atan(t2+h2)-atan(h2-
1.676331)+0.00391838*pow(h2, 1.5)*atan(0.023101*h2)-4.686035;
char tbuExterna[10];
dtostrf(tbuOut,4, 1, tbuExterna);

float tbuIn = t*atan(0.151977*pow((h+8.313659),0.5))+atan(t+h)-atan(h-
1.676331)+0.00391838*pow(h2, 1.5)*atan(0.023101*h)-4.686035;
char tbuInterna[10];
dtostrf(tbuIn,4, 1, tbuInterna);

float es = 6.112*exp((17.67*t)/(t+243.5));
float e = (es*h)/100;
float td = (243.5*log(e/6.112))/(17.67-log(e/6.112));
char tempDew[10];
dtostrf(td,4, 1, tempDew);

temperature_time = millis();
client.publish("reuniao/temperatura",temperaturaInterna);
client.publish("reuniao/umidade",umidadeInterna);
client.publish("reuniao/tbu",tbuInterna);
client.publish("reuniao/tempdew",tempDew);
client.publish("externa/temperatura",temperaturaExterna);
client.publish("externa/umidade",umidadeExterna);
client.publish("externa/tbu",tbuExterna);

}

```

```
if (time_passed > 300000)
{
  digitalWrite(resetArduino,LOW);
}
else
  digitalWrite(resetArduino,HIGH);
}
//LOOP-----
```