



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E AUTOMAÇÃO
CURSO DE ENGENHARIA DE COMPUTAÇÃO

JAIME CRISTALINO JALES DANTAS

SISTEMA MICROCONTROLADO DE MONITORAMENTO E CONTROLE DE
AMBIENTE PARA DATA CENTERS

NATAL

2017

JAIME CRISTALINO JALES DANTAS

**SISTEMA MICROCONTROLADO DE MONITORAMENTO E CONTROLE DE
AMBIENTE PARA DATA CENTERS**

Monografia apresentada ao Curso de Engenharia de Computação da Universidade Federal do Rio Grande do Norte, como requisito obrigatório para obtenção do título de Engenheiro de Computação.

ORIENTADOR: Professor Dr. Carlos Manuel Dias Viegas

NATAL

2017

JAIME CRISTALINO JALES DANTAS

**SISTEMA MICROCONTROLADO DE MONITORAMENTO E CONTROLE DE
AMBIENTE PARA DATA CENTERS**

Monografia apresentada ao Curso de Engenharia de Computação da Universidade Federal do Rio Grande do Norte, como requisito obrigatório para obtenção do título de Engenheiro de Computação.

Aprovado em **xx de fevereiro de 2017**.

Banca Examinadora

Prof. Dr. Carlos Manuel Dias Viegas
Departamento de Engenharia de Computação e Automação
Universidade Federal do Rio Grande do Norte

Prof. Dr. Luiz Affonso H. Guedes de Oliveira
Departamento de Engenharia de Computação e Automação
Universidade Federal do Rio Grande do Norte

Prof. Dr. Ivanovitch Medeiros Dantas da Silva
Instituto Metrópole Digital
Universidade Federal do Rio Grande do Norte

“Em algum lugar, algo incrível está esperando para ser descoberto”.
(Carl Sagan)

Dedico este trabalho aos meus pais,
Armando Jales e Júlia Dantas,
pelo incentivo que recebi durante
toda minha vida para me dedicar
a minha carreira acadêmica.

Agradecimentos

Muitos contribuíram, de diferentes formas, para que este trabalho se realizasse. Agradeço primeiramente aos meus pais pelo apoio que recebi durante minhas decisões e pelo suporte e incentivo que me deram para que eu obtivesse sucesso na minha carreira acadêmica e profissional.

Ao meu orientador, Prof. Dr. Carlos Viegas, pela paciência, esforço e disponibilidade no decorrer dos últimos dois anos e ao Prof. Dr. Diogo Pedrosa pelo suporte que recebi por parte da coordenação do curso.

Ao Engenheiro Kleydson Wilbert e a todos colegas de trabalho do PoP-RN pelo apoio e suporte que recebi durante meu estágio e também pelo incentivo que recebi para desenvolver este trabalho durante meu estágio.

Aos meus irmãos, pelo auxílio prestado no desenvolvimento deste trabalho. A minha família, pelo apoio durante essa jornada.

Ao meu amigo Alexandre Luz, pelo conhecimento que me foi passado ao longo de quase dois anos.

À CAPES, pela oportunidade de intercâmbio durante um ano e meio no Canadá e aos demais colegas de graduação, pelas críticas e sugestões.

Resumo

Este trabalho propõe um sistema composto por uma arquitetura de hardware embarcada e um software multiplataforma capaz de realizar o monitoramento e controle do ambiente de data centers. O projeto baseia-se em um microcontrolador AVR ATmega328p que se comporta como um agente SNMP. Foram projetadas duas fontes de tensão sendo uma fonte linear e outra do tipo chaveada para alimentar o circuito de controle e equipamentos externos. A arquitetura de hardware desenvolvida é capaz de monitorar a temperatura de vários sensores dispersos no data center e de controlar a abertura da porta do rack de comunicação através de uma fechadura magnética. Além disso, o sistema embarcado foi projetado para realizar o acionamento remoto de condicionadores de ar localizados no data center. Em adição ao hardware criado, foi desenvolvido um software multiplataforma capaz de realizar o gerenciamento, monitoramento e controle simultâneo de todos os dispositivos desenvolvidos. O sistema proposto é capaz de diagnosticar falhas no data center e nos dispositivos a partir da análise em tempo real dos dados recebidos pelas estações de monitoramento criadas. A comunicação entre os dispositivos e entre o sistema é realizada através do protocolo de gerenciamento de redes SNMP, o que torna o projeto inovador e promissor para aplicações industriais e de monitoramento de ambientes de redes de comunicação. De modo a validar a proposta, o trabalho apresenta resultados experimentais de aplicações reais onde o dispositivo desenvolvido foi empregado e traz testes de monitoramento usando softwares usados em ambientes de gerenciamento de redes. Os resultados obtidos são muito promissores e o objetivo é que esta proposta resulte na criação de um equipamento comercial de baixo custo que seja usado para realizar o monitoramento de ambientes de diferentes aplicações.

Palavras-chave: Sistemas Embarcados, Monitoramento de Ambiente de Data Centers, Internet das Coisas, Sistemas SNMP, Gerência de Redes.

Abstract

In this work, we propose a system composed of an embedded hardware architecture and a multiplatform software capable of monitoring and controlling a data center environment. The design is based on an ATmega328p AVR microcontroller that behaves as a SNMP agent. Two different power suppliers were designed, one is a linear voltage source and the other is a step-down converter for powering the control circuit and the external equipment. The developed hardware architecture is able to monitor the temperature of several scattered sensors in the data center, and to control the opening of the communication rack door through a magnetic lock. Also, the embedded system is designed to perform the remote actuation of air conditioners located in the data center. In addition to the hardware created, a multiplatform software was developed capable of performing the management, monitoring and simultaneous control of all the developed devices. The proposed system is able to diagnose failures in the data center and on the devices projected by real-time analysis of the data received from the monitoring stations created. The communication between the devices and the software is performed through the SNMP network management protocol, which makes this project innovating and promising for industrial applications and for monitoring network environments. In order to validate our proposal, we present experimental results of real applications where the developed devices were employed and we show monitoring tests using one of the most used software in the network management environment. The obtained results are very promising and the goal is that this work results in the development of a low cost commercial equipment that would be used to perform the monitoring of environments in different applications.

Keywords: Embedded Systems, Data Center Environment Monitoring, Internet of Things, SNMP Systems, Network Management.

Sumário

Sumário	i
Lista de Figuras	iii
Lista de Tabelas	v
Lista de Siglas e Abreviaturas	vi
1 Introdução	1
1.1 Objetivos	1
1.2 Motivação.....	2
1.3 Contribuições.....	2
1.4 Estrutura do Trabalho	3
2 Sistemas Embarcados	4
2.1 Microcontrolador	5
2.2 Shield Ethernet.....	7
2.3 Sensores de Temperatura	7
2.3.1 Termistor	8
2.3.2 DHT11	9
2.3.3 DHT22	9
2.4 Display LCD	10
2.5 Fechadura Magnética	11
2.6 Ar-Condicionado	11
2.7 Relé de Estado Sólido.....	11
3 Protocolo de Gerenciamento de Redes SNMP	13
3.1 Introdução	13
3.2 Pacote SNMP	14
3.3 Árvore de identificadores.....	15
4 Trabalhos relacionados	18
5 Sistema Microcontrolado	20
5.1 Funcionamento.....	20
5.2 Pinagem	23

6 Circuito Eletrônico	25
6.1 Circuito de Alimentação.....	25
6.1.1 Fonte Principal de 12V.....	25
6.1.2 Fonte Secundária de 5V.....	26
6.2 Circuito de Controle.....	28
6.3 Placa de Circuito Impresso.....	30
7 Sistema de Controle e Monitoramento Java	32
7.1 Linguagem de Programação Java.....	32
7.2 NetBeans	32
7.3 Bibliotecas	33
7.3.1 API SNMP para Java SNMP4J	33
7.3.2 API Gráfica para Java JfreeChart.....	33
7.4 Especificações do Sistema.....	33
7.5 Diagramas UML	34
7.5.1 Diagrama de Caso de Uso.....	35
7.5.2 Diagrama de Atividades	35
7.6 Funcionamento do Sistema	37
7.6.1 Módulo de Monitoramento	37
7.6.2 Módulo de Controle	43
7.6.3 Módulo de Gerenciamento	45
8 Resultados Obtidos	48
8.1 Equipamento desenvolvido	48
8.1.1 Análise Elétrica.....	53
8.2 Monitoramento e Controle com o Software DC Control	54
8.3 Monitoramento e Controle por um Smartphone	57
8.4 Monitoramento com o Software Zabbix	58
9 Conclusões	61
9.1 Trabalhos Atuais e Futuros	61
Referências Bibliográficas	62
Apêndices	66
APÊNDICE A – Projeto Eletrônico do Equipamento.....	66

Lista de Figuras

Figura 2.1: Pinagem do ATmega328p (Atmel Corporation, 2015)	5
Figura 2.2: Plataforma de desenvolvimento Arduino Uno (Arduino, 2017)	6
Figura 2.3: Shield Ethernet W5100 (Arduino, 2016)	7
Figura 2.4: Termistor NTC (Waltronica, 2017).....	8
Figura 2.5: Sensor DHT11 (Aosong Electronics, 2016).....	9
Figura 2.6: Sensor DHT22 (Aosong Electronics, 2016).....	10
Figura 2.7: Display LCD (FelipeFlop, 2017)	10
Figura 2.8: Fechadura Magnética (Ipec, 2016)	11
Figura 2.9: Relé de Estado Sólido (Fotek, 2016)	12
Figura 3.1: Pacote SNMP	14
Figura 3.2: Captura do Pacote SNMP.....	15
Figura 3.3: Árvore de Identificadores de Objetos (Santos et al., 2015)	15
Figura 3.4: Captura de Comunicação SNMP.....	16
Figura 5.1: Arquitetura do Sistema de Hardware.....	20
Figura 5.2: Fluxograma de Funcionamento do Sistema Embarcado.....	22
Figura 5.3: Ligação do Microcontrolador	24
Figura 6.1: Circuito Eletrônico Fonte Linear.....	26
Figura 6.2: Conversor de Tensão Buck.....	27
Figura 6.3: Circuito de Controle.....	29
Figura 6.4: Parte Frontal da PCB	30
Figura 6.5: Parte Inferior da PCB	31
Figura 7.1: Diagrama de Uso Sistema DC Control.....	35
Figura 7.2: Diagrama de Atividade de abertura de porta.....	36
Figura 7.3: Diagrama de Atividade de Acionamento do Ar-Condicionado	37
Figura 7.4: Tela de Login	38
Figura 7.5: Tela de Escolha das Estações Ativas.....	39
Figura 7.6: Threads do Sistema	39
Figura 7.7: Threads do Sistema.....	40
Figura 7.8: Tela de Seleção de Sensores.....	41
Figura 7.9: Tela dos Gráficos.....	42
Figura 7.10: Tela de Configuração da Temperatura	43
Figura 7.11: Tela de Controle da Estação	44
Figura 7.12: Avisos de Ar-Condicionado Desligado e Porta Fechada	44
Figura 7.13: Tela de Temporizador do Acionamento do Ar-Condicionado.....	45
Figura 7.14: Tela de Edição das Estações.....	46
Figura 7.15: Tela de Diagnósticos de Falhas	47
Figura 7.16: Retorno do Teste de Comunicação SNMP	47
Figura 8.1: Equipamento Projetado.....	48

Figura 8.2: Equipamento Desenvolvido	49
Figura 8.3: Placa Eletrônica Montada	50
Figura 8.4: Placa de Circuito Impresso Desenvolvida.....	50
Figura 8.5: Projeto em Protoboard.....	51
Figura 8.6: Projeto em Desenvolvimento no PoP-RN	52
Figura 8.7: Sensor de Temperatura Desenvolvido	52
Figura 8.8: Gráfico de Saída da Fonte Linear 12V	53
Figura 8.9: Gráfico da Saída da Fonte Buck 5V	54
Figura 8.10: Rede Criada para Testes	55
Figura 8.11: Teste de Temperatura das Estações	56
Figura 8.12: Teste de Temperatura Estação TVU.....	56
Figura 8.13: Monitoramento da Temperatura pelo Smartphone.....	57
Figura 8.14: Mapa de Rede com o Zabbix	58
Figura 8.15: Variáveis Gerenciáveis com o Zabbix	59
Figura 8.16: Tela de Gerenciamento Zabbix.....	59
Figura 8.17: Mensagem do Envio de Email do Zabbix	60

Lista de Tabelas

Tabela 2.1: Características do ATmega328p	6
Tabela 2.2: Constantes da Equação de Steinhart-Hart.....	8
Tabela 2.3: Formato Mensagem DHT11.....	9
Tabela 3.1: Identificadores Privados do Equipamento.....	16
Tabela 5.1: Pinagem do Microcontrolador.....	24

Lista de Siglas e Abreviaturas

AC	Alternating Current
ASCII	American Standard Code for Information Interchange
C	Celsius
CI	Circuito Integrado
CMOS	Complementary Metal–Oxide–Semiconductor
CPU	Central Processing Unit
DC	Direct Current
E/S	Estrada/Saída
EEPROM	Electrically Erasable Programmable Read-Only Memory
gigaPOP	Gigabit Point of Presence
ICSP	In-Circuit Serial Programming
IDE	Integrated Development Environment
IP	Internet Protocol
LCD	Liquid Crystal Display
LED	Light-Emitting Diode
MIPS	Million Instructions per Second
PCB	Printed Circuit Board
PoP	Point of Presence
PWM	Pulse Width Modulation
RFC	Request for Comments
RISC	Reduced Instruction Set Computer
ROM	Read-Only Memory
SD	Storage Device
SMS	Short Message Service
SNMP	Simple Network Management Protocol
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory
TCP	Transmission Control Protocol
TTL	Transistor-Transistor Logic
UML	Unified Modeling Language
W	Watt

Capítulo 1

1 Introdução

Sistemas de monitoramento de ambiente são usados em larga escala ao redor do mundo. Desde de monitoramento de ambientes industriais até mesmo para previsões meteorológicas se tem usado sensores e dispositivos eletrônicos para monitorar o ambiente. De acordo com Ferdoush & Li (2014), uma nova era de desenvolvimento de hardware e software de código aberto está surgindo e irá transformar o cenário mundial em alguns anos. Além disso, a Internet das Coisas (IoT) está cada vez mais presente em nosso cotidiano, conectando dispositivos de diferentes funções através da Internet.

O monitoramento dos recursos de sistemas embarcados por um protocolo de gerenciamento de redes é muito importante pois seu hardware é limitado. De acordo com Sehgal et al. (2012), a implementação do protocolo de gerenciamento de redes SNMP em microcontroladores da família AVR do fabricante Atmel é possível quando se adotam técnicas de simplificações na estrutura de funcionamento do protocolo criando uma versão mais simplificada e menos complexa que atende a todas as necessidades do problema.

É nesse cenário que é proposto um sistema embarcado capaz não somente de monitorar o ambiente mas também de realizar o controle de equipamentos. O sistema proposto é um conjunto composto por uma arquitetura de hardware embarcada, um sistema eletrônico e uma aplicação de software desenvolvida para solucionar um problema real enfrentado por diversas empresas que possuem grandes data centers ao redor do mundo.

Este trabalho foi desenvolvido ao longo de seis meses em parceria com o PoP-RN da Rede Nacional de Pesquisa (RNP).

1.1 Objetivos

O objetivo principal deste trabalho é solucionar um problema existente no data center e nos gigaPOPs da empresa PoP-RN. Um gigaPOP é um ponto de acesso de internet individual de alta velocidade geralmente mantidos por universidades e órgãos governamentais. Os gigaPOPs geralmente possuem velocidades de conexão iguais ou superiores a 1 Gigabit por segundo. O sistema proposto foi desenvolvido para realizar o monitoramento da temperatura do ambiente, acionar a abertura das portas dos racks

de comunicação localizados no data center e nos gigaPOPs, e por fim realizar o acionamento do ar-condicionado dos gigaPOPs de forma remota.

Ao todo são 10 gigaPOPs localizados em diferentes cidades do estado do Rio Grande do Norte. Grande parte dos gigaPOPs possuem condicionadores de ar reservas que são acionados manualmente na ocorrência de falhas no ar-condicionado principal. O sistema proposto foi idealizado para realizar o revezamento dos condicionadores de ar, assim como também o acionamento do ar-condicionado secundário caso a temperatura do ambiente esteja muito elevada.

Além disso, todo sistema desenvolvido usa o protocolo de gerenciamento de redes SNMP para comunicação, facilitando assim a operação e manuseio por qualquer administrador de rede que possua a documentação adequada do sistema. Em adição, a solução desenvolvida possui baixo custo de fabricação.

1.2 Motivação

A possibilidade da criação de um sistema embarcado baseado em hardware e software de código aberto com comunicação SNMP para realizar o monitoramento e controle do ambiente, mostra-se muito promissor em um cenário onde o custo das tecnologias relacionadas ao tema tem caído drasticamente nos últimos anos.

As soluções existentes no mercado para atender este nicho são todas de tecnologias estrangeiras e de elevado valor aquisitivo. O sistema proposto neste trabalho possui um custo de mais de 10 vezes inferior do que a solução mais em conta no mercado norte americano. O propósito principal deste projeto é a criação de um equipamento comercial que atenda todas as necessidades existentes na área de monitoramento de ambiente para data centers.

1.3 Contribuições

Dentre as principais contribuições deste trabalho, podem ser destacadas:

- O desenvolvimento de um sistema embarcado baseado no microcontrolador ATmega328p com comunicação SNMP.
- A criação de uma arquitetura de hardware embarcada baseada na eletrônica analógica e digital, sensoriamento e acionamento de sistemas.
- O projeto de uma software de monitoramento e controle de ambiente de data centers dos gigaPOPs baseado na linguagem Java.
- A configuração de um ambiente de alarmes e avisos por email usando o software de gerenciamento de redes Zabbix.

1.4 Estrutura do Trabalho

O presente trabalho está dividido em 8 capítulos, além desta introdução. O capítulo 2 introduz brevemente os principais conceitos dos dispositivos eletrônicos que serão usados no sistema de hardware desenvolvido e o capítulo 3 traz uma revisão dos conceitos de gerenciamento de redes com o protocolo SNMP. Já o capítulo 4, por sua vez, traz uma revisão da literatura a partir das estratégias previamente propostas e relacionadas a este trabalho. No capítulo 5 é mostrado o funcionamento do sistema embarcado. No capítulo 6 o sistema embarcado e eletrônico são propostos. O capítulo 7 mostra o desenvolvimento do software de gerenciamento e controle. O capítulo 8 mostra os resultados obtidos com o sistema proposto. Por fim, o capítulo 9 conclui o trabalho.

Capítulo 2

2 Sistemas Embarcados

Nos dias atuais, estamos cada vez mais dependentes de sistemas embarcados. Tecnologias embarcadas estão presentes em nossas atividades diárias e muitas vezes não as percebemos. Eletrodomésticos como televisão, aparelho de som, e micro-ondas são construídos baseados em sistemas embarcados. Além disso, nossos carros também usam diversos sistemas tecnológicos que são construídos com base em sistemas embarcados.

Segundo Shibu (2009), a revolução computacional teve início com computadores de propósito geral. Mas tarde percebeu-se que computação de uso geral não era suficiente para a nova era computacional chamada de computação embarcada pois sistemas embarcados necessitavam de características especiais com relação a consumo energético, memória, processamento, etc.

No entanto, sistemas embarcados já estavam presentes antes da revolução computacional e eram baseados nos antigos tubos de vácuo e tecnologias de transistores com algoritmos desenvolvidos em linguagem de baixo nível. Os sistemas embarcados podem ser classificados como:

- *Sistemas embarcados de pequena escala:* São sistemas simples usados em aplicações onde performance não é primordial. Geralmente são usados microcontroladores de 8 ou 16 bits.
- *Sistemas embarcados de média escala:* São sistemas com determinada complexidade de hardware e software. Geralmente são usados microcontroladores de 16 ou 32 bits ou até mesmo processadores digitais de sinais.
- *Sistemas embarcados de larga escala:* São sistemas que envolvem uma complexidade de firmware muito elevada e demandam alta performance. Geralmente são usados processadores de 32 ou 64 bits de vários núcleos de processamento. Aplicações de processamento de sinais em tempo real são exemplos de sistemas de embarcados de larga escala.

Os propósitos de sistemas embarcados são diversas. No entanto, as principais aplicações envolvem armazenamento de dados, comunicação, processamento de dados, monitoramento e controle.

O sistema embarcado proposto neste trabalho está enquadrado na categoria de monitoramento e controle. Muitos dos sistemas desta categoria estão presentes na área médica, onde equipamentos são usados para determinar variáveis de monitoramento de

um paciente. Podemos citar como exemplo o aparelho de eletro cardiógrafo que monitora os batimentos cardíacos de uma pessoa através de sensores. Um sistema de refrigeração de uma sala é um exemplo de sistemas embarcados de controle pois a temperatura do ambiente é controlada através do microcontrolador que processa o valor do sensor de temperatura e aciona ou desliga o sistema de ar-condicionado.

Neste capítulo serão apresentados e discutidos todos os componentes e dispositivos usados no desenvolvimento do hardware do sistema embarcado desenvolvido.

2.1 Microcontrolador

Microcontroladores são circuitos integrados que contém um CPU, memória RAM, registradores de uso geral, memória ROM e Flash, memória para programação do firmware e de armazenamento, temporizadores, unidades de controles de interrupção e portas de entrada e saída (Shibu, 2009). Em outras palavras, microcontroladores podem ser considerados como um arranjo de vários microprocessadores.

O microcontrolador usado foi o ATmega328p, que é um microcontrolador CMOS de 8 bits de baixo consumo energético baseado na arquitetura RISC. O ATmega328p atinge uma *throughput* de até 1MIPS por MHz (Atmel Corporation, 2015).

A figura 2.1 mostra a pinagem do Atmega328p. Foi escolhido usar a plataforma de desenvolvimento Arduino Uno e as suas características são mostradas na tabela 2.1. A escolha foi devido a facilidade de gravação de firmware no microcontrolador, além do baixo custo da placa Arduino.

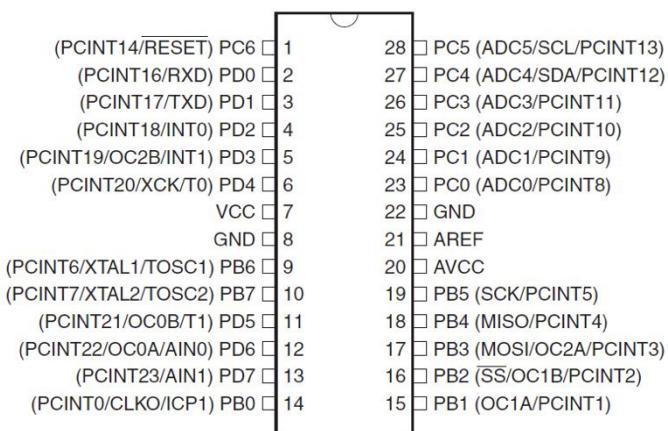


Figura 2.1: Pinagem do ATmega328p (Atmel Corporation, 2015)

Tabela 2.1: Características do ATmega328p

Microcontrolador	ATmega328p
Tensão Operacional	5V
Pinos Digitais E/S	14
Pinos PWM	6
Pinos Analógicos	6
Corrente DC por Pino E/S	20 mA
Corrente DC para o Pino 3.3V	50 mA
Memória Flash	32 KB
SRAM	2 KB
EEPROM	1 KB
Relógio Interno	16 MHz
LED_BUILTIN	13
Comprimento	68,6 mm
Largura	53,4 mm
Peso	25 g

O Arduino Uno já possui todos os componentes necessário para o uso do microcontrolador ATmega328p. Além disso, ele já possui um CI que converte a comunicação USB para serial e permite a gravação de novos firmwares diretamente via USB. Este conversor é baseado no ATmega16U2, porém em algumas versões paralelas são usados outros conversores. Uma outra característica da plataforma Arduino é a inclusão de uma comunicação ICSP que é usado para comunicação com o Shield Ethernet. A programação do microcontrolador foi feita em linguagem de baixo nível C usando o software Arduino IDE 1.6.12. A figura 2.2 mostra o Arduino Uno que foi usado.



Figura 2.2: Plataforma de desenvolvimento Arduino Uno (Arduino, 2017)

2.2 Shield Ethernet

O Shield Ethernet usado é baseado no WIZnet W5100, um chip de internet 10/100 controlador Ethernet desenvolvido para aplicações embarcadas que possui fácil integração, estabilidade e performance. O W5100 é IEEE 802.3 10BASE-T e 802.3U 100BASE-TX compatível.

Além disso, o W5100 inclui uma pinha TCP/IP e tem suporte para conexões TCP, UDP, IPv4, ICMP, ARP, IGMP e PPPoE. Possui um buffer interno de 16 Kbytes para transmissão de dados. A programação é de fácil implementação e a Arduino disponibiliza uma biblioteca oficial para uso grátis. A figura 2.3 mostra o Shield Ethernet.



Figura 2.3: Shield Ethernet W5100 (Arduino, 2016)

O Shield Ethernet também possui uma conexão para cartão SD que pode ser usada para armazenamento de dados. A comunicação entre o Arduino e o Shield Ethernet é feita pelo protocolo SPI usando os pinos ICSP do Arduino. A comunicação entre estes dispositivos também faz uso dos pinos digitais do Arduino PIN 13, 12, 11 e 10.

2.3 Sensores de Temperatura

O sistema desenvolvido usa vários tipos de sensores dependendo das necessidades de cada aplicação. No entanto, por padrão, todos equipamentos são equipados com um sensor de temperatura interna do tipo termistor e um sensor externo de temperatura e umidade modelos DHT11 ou DHT22. Outros sensores podem ser adicionados ao sistema de acordo com a necessidade de cada cenário.

2.3.1 Termistor

Cada equipamento possui um sensor de temperatura acoplando na placa do circuito. O valor da temperatura interna do equipamento é usada como parâmetro de segurança e verificação de falhas pelo sistema DC Control. O termistor usando é do tipo NTC(Negative Temperature Coefficient) onde a resistência diminui a medida que a temperatura aumenta. O termistor é capaz de ler temperaturas de -40° até 125° C.

Assumindo uma aproximação linear entre resistência e temperatura, temos:

$$\Delta R = k\Delta T \quad (2.1)$$

Onde ΔR é a variação da resistência, ΔT a variação de temperatura e k o coeficiente de primeira ordem de resistência. Na prática, a aproximação linear da equação 2.1 não funciona para pequenos intervalos de temperatura. Para se obter uma temperatura mais exata, a aproximação de terceira ordem de Steinhart-Hart é aplicada. A equação 2.2 mostra a aproximação de terceira ordem onde podemos notar a presença de três constantes a , b e c .

$$\frac{1}{T} = a + b \ln(R) + c(\ln(R))^3 \quad (2.2)$$

Essas constantes são disponibilizadas pelo fabricante do termistor usando. Segundo Malesevic & Stupic (2011), um resistor de 10 kΩ deve ser usado em série com o termistor. A tabela 2.2 mostra os parâmetros da equação 2.2.

Tabela 2.2: Constantes da Equação de Steinhart-Hart

Parâmetro	Valor
a	0.001129148
b	0.000234125
c	0.0000000876741

A figura 2.4 mostra o termistor usado. Uma das maiores vantagens desse tipo de sensor é o baixo custo. Enquanto outros tipos de sensores custam dezenas de reais, um termistor custa apenas alguns centavos de real.



Figura 2.4: Termistor NTC (Waltronica, 2017)

2.3.2 DHT11

O sensor DHT11 mede a temperatura e umidade e tem uma saída digital de dados onde o sinal de temperatura e umidade são transmitidos. O sensor é composto por um componente úmido para medição da umidade e um sensor termistor NTC conectados a um microcontrolador de 8 bits que é responsável pela comunicação com o microcontrolador ATmega328p.

A principal vantagem do DHT11 é o baixo custo, o colocando na categoria de sensores mais baratos existentes para sistemas embarcados. Como consequência do baixo custo, o sensor DHT11 possui uma resolução de medição muito pequena. A precisão para medição da temperatura é de $\pm 2^\circ \text{C}$ para uma temperatura de 25°C com resolução de $\pm 1^\circ$. Para a medição da umidade a imprecisão é ainda maior, de $\pm 5\%$ em temperatura ambiente. Além disso, a taxa de atualização do sensor é de aproximadamente 3 segundos e o range de temperatura é de apenas 50 graus, variando de 0° até 50°C (Aosong Electronics, 2016).

Um resistor de *pull-up* de $10 \text{k}\Omega$ deve ser usado na linha de dados do sensor para tamanhos pequenos de cabos. A comunicação usada pelo microcontrolador presente no sensor é do tipo *single bus* na estrutura mestre-escravo. O formato da mensagem é apresentada na tabela 2.3.

Tabela 2.3: Formato Mensagem DHT11

Humidade Inteiro	Humidade Decimal	Temperatura Inteiro	Temperatura Fracionário	Paridade
8 bit	8 bit	8 bit	8 bit	8 bit

O fabricante do sensor disponibiliza uma biblioteca gratuita para uso em microcontroladores AVR da Atmel. A figura 2.5 mostra o sensor DHT11.

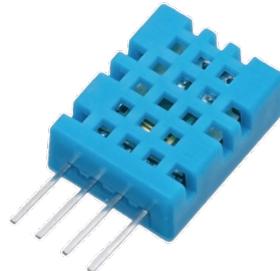


Figura 2.5: Sensor DHT11 (Aosong Electronics, 2016)

2.3.3 DHT22

O Sensor DHT22 é uma versão mais robusta do sensor DHT11. O DHT22 possui uma alta confiabilidade devido a alta resolução do sensor. Este sensor tem a capacidade de comunicação sem perdas por um cabo de até 20 metros de comprimento. Como

consequência, o custo do DHT22 é muito superior ao do DHT11, chegando a custar até 4 vezes mais caro do que seu concorrente. Portanto, apenas alguns dos dispositivos usam o sensor DHT22, enquanto outra parte utiliza o DHT11.

A resolução de medição da temperatura do DHT11 é de $\pm 0,1^{\circ}\text{C}$ com uma precisão de $\pm 0,5^{\circ}\text{C}$. Já para medição de umidade, a resolução é de 2 % para temperatura ambiente. Além disso, o sensor tem um range que varia de -40° até 80° C (Aosong Electronics, 2016).

A figura 2.6 mostra o sensor DHT22.

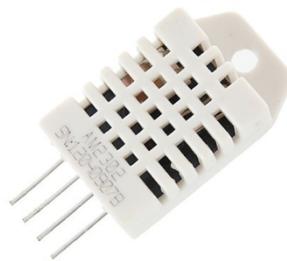


Figura 2.6: Sensor DHT22 (Aosong Electronics, 2016)

2.4 Display LCD

O equipamento desenvolvido possui um display LCD para mostrar ao operador informações úteis. O display usado é o Alphanumeric LCD Display 16x2. Este display é capaz de mostrar 224 diferentes caracteres e símbolos, e funciona com alimentação de 5V. O display possui 16 pinos de entrada que são ligados a placa do circuito eletrônico através de um cabo do tipo flat (Revolution Education, 2016).

O display mostra até 16 caracteres por linha, e possui duas linhas. Cada caractere é mostrado como uma matriz de pixel de 5x7. Este display possui dois registradores responsáveis pelo armazenamento da instrução de comando, que significa a tarefa a ser realizada como limpar o display ou definir a posição do cursor, e pelo armazenamento do dado no LCD. Esse dado é o código ASCII de cada caractere mostrado no LCD.

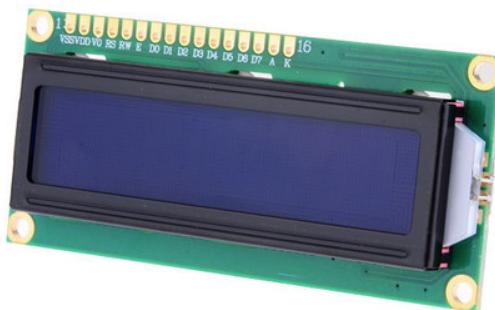


Figura 2.7: Display LCD (FelipeFlop, 2017)

2.5 Fechadura Magnética

A fechadura usada no rack de telecomunicações é do tipo magnética de 150 kgf. A fechadura magnética é um dispositivo que tem como princípio de funcionamento a força de tração eletromagnética. Sua função se dá através do contato das superfícies do blanque e do eletroímã resultando numa forma de tração, de aproximadamente 150 kgf (Ipec, 2016).

A alimentação da fechadura é de 12V DC e o consumo de corrente é de 400 mA. A figura 2.8 mostra a fechadura usada.



Figura 2.8: Fechadura Magnética (Ipec, 2016)

2.6 Ar-Condicionado

O acionamento do ar-condicionado é feito através do ligamento da alimentação elétrica diretamente do aparelho. Todos os condicionadores de ar usados são do tipo Split de potência média de 18.000 BTU/h (5.274 W) com um consumo de corrente em regime permanente de 8,4A em 220V a 60Hz.

Para que o acionamento do ar-condicionado fosse possível, foi usando um recurso presente nos condicionadores de ar modernos do tipo Split chamado de *auto start*. Nessa função, quando reestabelecido a alimentação do aparelho, o mesmo entrará em funcionamento e fará o resfriamento de acordo com as condições previamente estabelecidas antes do desligamento da alimentação do aparelho.

2.7 Relé de Estado Sólido

O acionamento do ar-condicionado é feito através de um relé de estado sólido, também conhecido com Solid-State relay (SSR). Um SSR é um dispositivo eletrônico

de chaveamento acionado por uma pequena tensão aplicada em um circuito controlador. O chaveamento é realizado por um triac de alta potência que é capaz de suportar centenas e amperes de corrente AC entre seu coletor e emissor. Os SSR possuem inúmeras vantagens com relação a relés mecânicos pois não possuem desgaste de contatos mecânicos, não geram ruídos magnéticos no sistema e apesar do tamanho reduzido são capazes de realizar o chaveamento de grandes cargas.

Um acoplador ótico é usado para ativar a condução entre emissor e coletor do triac. O relé de estado sólido que foi usado é o FOTECK SSR-40 DA que é capaz de realizar o chaveamento de até 40A em 380V AC e possui entrada de tensão de 2-32V DC. A figura 2.9 mostra o SSR usando.



Figura 2.9: Relé de Estado Sólido (Fotek, 2016)

Capítulo 3

3 Protocolo de Gerenciamento de Redes SNMP

Simple Network Management Protocol (SNMP) é o protocolo de redes mais usado para gerenciamento de redes. Ele é usado para coletar informações e para definir configurações de dispositivos de redes como servidores, impressoras, hubs, switches e roteadores usando o protocolo de internet IP. Neste capítulo, serão discutidos o funcionamento deste protocolo e a modelagem que foi realizada no sistema proposto.

3.1 Introdução

Grandes redes com centenas ou mesmo milhares de nós são muito difíceis de gerenciar sem que haja um operador dedicado para monitorar cada dispositivo. O protocolo SNMP, que é largamente usado em redes de área local (LANs), permite o monitoramento dos nós da rede por apenas um gerente de monitoramento chamado gerente SNMP. Podemos monitorar dispositivos como servidores, computadores, impressoras, roteadores, bridges, hubs, serviços como o protocolo de configuração dinâmica de host (DHCP) e Windows Internet Name Service (WINS). Um software de gerenciamento SNMP é usado para monitorar os dispositivos de rede chamados de agente (Microsoft, 2003).

O protocolo SNMP foi desenvolvido em 1988 para prover o monitoramento de dispositivos de rede para o protocolo TCP/IP. No ano de 1990 o protocolo foi aprovado pela Internet Architecture Board (IAB) e tem sido largamente usado desde então (Internet Engineering Task Force, 1990).

De acordo com Santos et al. (2015), para realizar a gerência de redes é necessário obter informações de uma variedade considerável de equipamentos, dispostos em redes distintas. Para que essa diversidade não se constitua em um problema, há necessidade de padrões que definam um modelo de gerência baseada em gerente e agente.

As ações que podem ser realizadas no gerenciamento da rede estão definidas na MIB (Management Information Base) e a maneira que novas informações podem ser incorporadas ao modelo é padronizado pela SMI (Structure of Management Information).

Os equipamentos podem ser executados como um gerente ou um agente. O gerente é responsável pela coleta de informações que se deseja gerenciar e o agente é a entidade

instalada nos diversos nós da rede que se pretende gerenciar e que coleta informações no hardware e no SO para repassá-las para o gerente (Santos et al., 2015).

Quando o gerente envia uma mensagem para o agente, ele solicita o envio de um objeto de gerenciamento ou mesmo um conjunto deles. O conjunto de objetos gerenciados conhecidos por um agente são chamados de MIB. O nome do identificador de objeto (OID) é uma nome único que define o objeto que está sendo gerenciado.

3.2 Pacote SNMP

O protocolo SNMP possui 3 versões que foram definidas pelo Internet Engineering Task Force (IETF). A versão usada pelo sistema proposto é o SNMP versão 1 (SNMPv1) que é a versão inicial do protocolo. Ela é definida na RFC 1157 e sua segurança é baseada em comunidades, que são senhas baseadas em Strings. Isso permite que qualquer aplicação de captura de pacote possa acessar essa senha e ter acesso ao dispositivo que está sendo gerenciado. A figura 3.1 mostra a estrutura do pacote SNMP.

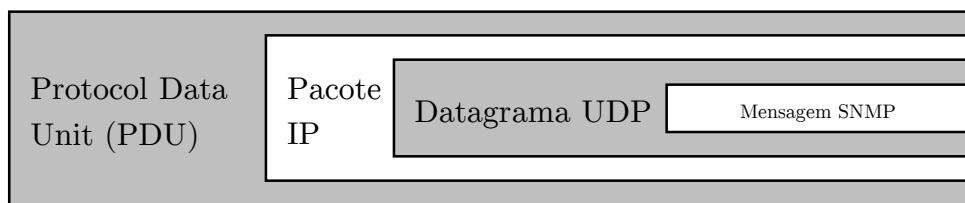


Figura 3.1: Pacote SNMP

O protocolo SNMP usa o User Datagram Protocol (UDP) para transportar o dado entre gerente e agente. O protocolo UDP, definido na RCF 768, roda sobre o protocolo TCP e não realiza conexão de ponto a ponto entre o agente e o gerente durante a comunicação. Esse aspecto do UDP o torna não confiável pois não existe identificação de falhas na transmissão. O SNMP usa uma simples sistema de controle por timeout para realizar a retransmissão e/ou identificar erros na comunicação (*Mauro & Kevin, 2005*).

A figura 3.2 mostra a captura de um pacote SNMP usando o software ¹Wireshark. Nela podemos notar que a comunidade SNMP é de fácil acesso, assim como todos o resto da mensagem.

¹ Wireshark é um software analisador de pacotes de redes distribuído gratuitamente. Este programa é de código aberto e está disponível para os SO MacOS, Linux e Windows.

```

▼ Simple Network Management Protocol
  version: version-1 (0)
  community: pop-rn
  ▼ data: get-response (2)
    ▼ get-response
      request-id: 725022438
      error-status: noError (0)
      error-index: 0
      ▼ variable-bindings: 1 item
        ▼ 1.3.6.1.2.1.6540.4.0: 4152202030207c20504f525441202030
          Object Name: 1.3.6.1.2.1.6540.4.0 (iso.3.6.1.2.1.6540.4.0)
          Value (OctetString): 4152202030207c20504f525441202030

```

Figura 3.2: Captura do Pacote SNMP

3.3 Árvore de identificadores

O identificador do objeto é formado com base em uma árvore de identificação de objetos. A figura 3.3 corresponde a uma parte da árvore de identificadores utilizada que é definida na RFC 1461.

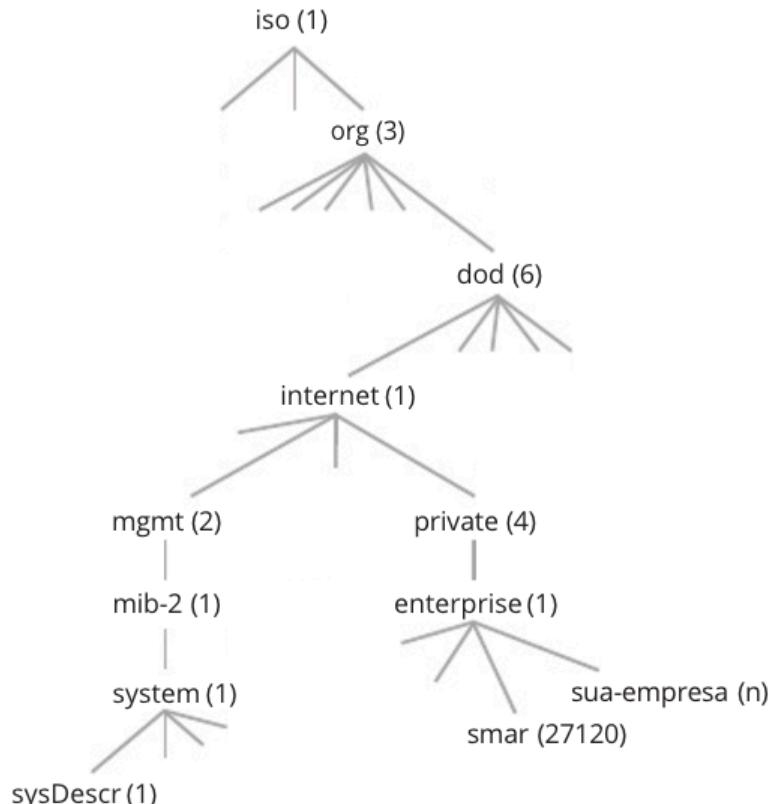


Figura 3.3: Árvore de Identificadores de Objetos (Santos et al., 2015)

A identificação do objeto gerenciado é feita pela junção dos números da árvore de OID separados por pontos. No grupo de *system* estão presentes uma lista de todas as operações gerais do sistema. Para o equipamento desenvolvido, foram implementados os seguintes objetos:

- **sysDescr**: representa a descrição da estação;
- **sysContact**: dados do responsável da estação;
- **sysLocation**: localização da estação;
- **sysUpTime**: Tempo de execução do sistema (último boot).

Sob a subárvore de objetos privados, localizados em iso.organization.dod.internet.private (1.3.6.1.4) foram colocados os identificadores para os objetos privados. A tabela 3.1 mostra todos os objetos privados criados e configurados no gerenciamento SNMP do dispositivo desenvolvido.

Tabela 3.1: Identificadores Privados do Equipamento

Objeto	OID	Função
ligaAR	1.3.6.1.4.1.9862.1.0	Liga Ar-Condicionado
desligaAR	1.3.6.1.4.1.9862.0.0	Desliga Ar-Condicionado
abriPORTA	1.3.6.1.4.1.3716.1.0	Abre Porta do Rack
fechaPORTA	1.3.6.1.4.1.3716.0.0	Fechar Porta do Rack
snmpTEMP1	1.3.6.1.4.1.6540.1.0	Temp. Sensor Externo
snmpTEMP2	1.3.6.1.4.1.6540.2.0	Temp. Sensor Equip.
snmpHUMIDADE	1.3.6.1.4.1.6540.3.0	Sensor Humidade
snmpOUTPUTS	1.3.6.1.4.1.6540.4.0	Valor das Saídas

O retorno do dado para todos os objetos privados é do tipo String. Todos os comandos usam a operação GetRequest que consiste em uma mensagem enviada do gerente ao agente solicitando um objeto em específico. A figura 3.4 mostra a captura de uma comunicação SNMP entre uma estação agente (IP 192.168.2.3) e o gerente de controle DC Control (IP 192.168.2.93) com o software Wireshark.

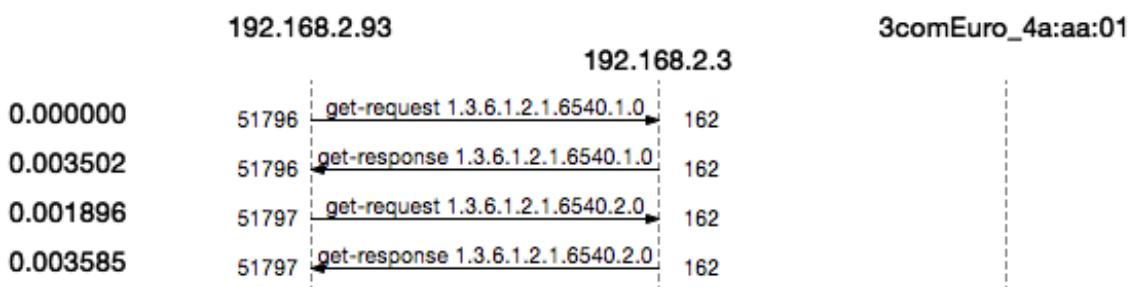


Figura 3.4: Captura de Comunicação SNMP

Nesta captura presente na figura 3.4 podemos observar em detalhes como ocorre a troca de informações entre gerente e agente SNMP. Primeiramente o gerente envia um comando GetRequest com o OID de requisição da temperatura do sensor externo

ao agente conectado na porta 162. Em seguida, o agente retorna para o gerente o valor da temperatura na mesma porta de recebimento da solicitação, ou seja, na porta 51796. Logo em seguida, o gerente faz outra solicitação ao agente e o processo se repete. Toda esta operação levou aproximadamente 3 milissegundos. Também podemos notar a presença de um switch da fabricante 3Com que realiza o encaminhamento dos pacotes entre o gerente e o agente.

Capítulo 4

4 Trabalhos relacionados

Alguns trabalhos na área de monitoramento de ambiente para data centers usando a plataforma de desenvolvimento Arduino e o protocolo de gerenciamento de redes SNMP foram desenvolvidos não somente do Brasil mas também ao redor do mundo. Um sistema de monitoramento de ambiente para data center é discutido por Fernández et al. (2014), onde foram usados o software de monitoramento de redes

¹Nagios integrado com uma aplicação que se comunicava com o Arduíno através de uma servidor dedicado. A solução apresentada consistia em uma rede de sensores de temperatura, humidade e temperatura dispersos em locais distintos dos racks de comunicação de um supercomputador. No entanto, este hardware não implementa o protocolo SNMP diretamente no microcontrolador, necessitando assim de uma servidor que se conecta ao Arduino por conexão serial e o interfaceamento da comunicação entre o Nagios e o Arduino é feito por um programa escrito na linguagem Python.

Uma outra solução é apresentada por Araújo & Junior (2013) em uma rede de sensoriamento SNMP usando a plataforma Arduino. Uma rede de sensores de temperatura foi desenvolvida e a comunicação entre servidor cliente é feita através de um Shield Ethernet conectado ao Arduino. Os dados são coletados via SNMP usando o Nagios. O sensoriamento implementado é executado em uma rede sem fio. No entanto, os autores não mostram como é feita a implementação em baixo nível do protocolo SNMP no Arduino.

Projetos envolvendo a Internet das Coisas (IoT) são cada vez mais comuns nos dias atuais. A abordagem adotada por Machado et al. (2015) mostra que o uso de redes de sensoriamento remoto sem fio com um hardware baseado no microcontrolador Arduino e com comunicação via SNMP tem inúmeras vantagens pois além de tornar a comunicação mais simplificada, permite a interação com softwares de gerenciamentos de redes.

No hardware proposto por Santos (2009) a arquitetura é baseada em um microcontrolador PIC onde é executado um servidor WEB que envia os valor do sensor de temperatura para o usuário através de um browser de Internet. Em sua proposta, foi

¹ Nagios é um software de gerenciamento de redes corporativas que trabalha com vários protocolos inclusive o SNMP. Ele possui uma licença gratuita e uma paga onde dispõe de suporte 24h. O Nagios possui um sistema de alarmes e avisos por email e por SMS.

desenvolvido um sistema capaz de monitorar e armazenar a temperatura de um data center.

A implementação de um sistema de monitoramento e controle de ambiente residencial é discutida por Romeiro & Costa (2016) onde é projetado um hardware que controla o acionamento das lâmpadas de um cômodo de uma casa através do protocolo SNMP com o software Zabbix. A programação dos software embarcado no Arduino foi baseada na biblioteca Agentuino e o acionamento das luzes é feito usando relés.

Em Born (2012), o autor propõe uma estação meteorológica usando o microcontrolador ATmega328p juntamente com um Shield Ethernet usando exclusivamente o protocolo de gerenciamento de redes SNMP. No entanto, é ressaltando que este microcontrolador não é robusto o bastante para executar todas as operações envolvidas na comunicação SNMP. Logo, o autor recomenda o uso de outros modelos como o ATmega2560 que possui uma quantidade de memória SRAM e Flash superior.

Em todos os projetos que usaram a biblioteca SNMP Agentuino foram apontados problemas relacionados a falta de memória do ATmega328p. A solução implementada foi o uso de um microcontrolador com mais memória RAM, que em alguns casos foram usados a plataforma de desenvolvimento Arduino Mega que é baseada no microcontrolador Atmega2560.

Capítulo 5

5 Sistema Microcontrolado

Sistemas microcontrolados, também chamados de sistemas eletrônicos embarcados, são sistemas computacionais eletrônicos ou eletromecânicos projetados para executar uma função em específica e são compostos por uma combinação de hardware e software (Shibu, 2009). O equipamento criado é composto por um microcontrolador ATmega328p da plataforma de desenvolvimento Arduino Uno. A figura 5.1 abaixo mostra a arquitetura de hardware do sistema.

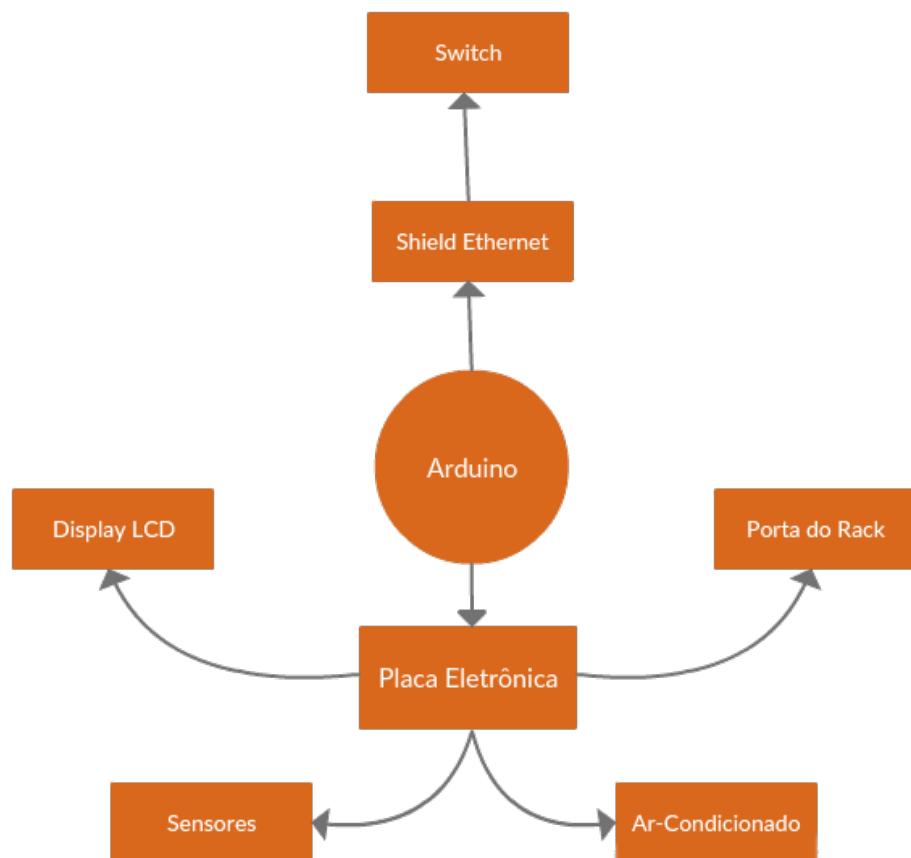


Figura 5.1: Arquitetura do Sistema de Hardware

5.1 Funcionamento

Devido a limitações de hardware, foi necessário realizar a programação dos sistema em baixo nível em C, adotando técnicas de otimização de uso de processamento de CPU. Além disso, por se tratar de um protótipo, foi escolhido usar a plataforma de desenvolvimento Arduino em vez de apenas o microcontrolador separadamente no modo *stand alone*.

O microcontrolador do Arduino é responsável por executar toda lógica do software projetado. A comunicação entre o Arduino e o Shield Ethernet é feito através do protocolo SPI. O Arduino se conecta a uma placa eletrônica que contém o circuito de alimentação, controle, e acionamento. O fluxograma da figura 5.2 mostra o funcionamento do software embarcado no Arduino.

Inicialmente, o sistema inicializa o display LCD, a comunicação SPI do Shield Ethernet, o sensor de temperatura DHT e o agente SNMP. Todas inicializações são realizadas na função *setup()* do Arduino, e é executada uma única vez. Além disso, é realizada a definição das entradas e saídas através do registrador DDR. Por fim, a saída do relé que ativa o ar-condicionado é colocada em nível baixo e o relé da abertura da porta do rack é colocado para nível alto. Essas configurações são ajustáveis de acordo com as necessidades de cada gigaPOP.

Na função do loop principal do Arduino *loop()* são executados três operações distintas. A primeira delas é a leitura dos sensores de temperatura. Esta leitura é realizada a cada 3 segundos, pois a taxa de atualização do sensor DHT11 está limitada a este intervalo de tempo. Não foi usando a função *delay()* do microcontrolador na programação do software pois esta função bloqueia o uso do microcontrolador no tempo que configuramos o delay. Para contornar essa situação, foi usado a função *millis()* do microcontrolador que retorna a quantidade de milissegundos desde da sua inicialização. Segundo as especificações do microcontrolador, a função *millis()* atinge o limite de overflow em aproximadamente 50 dias. (Arduino, 2017).

A função *millis()* também foi usada para calcular o tempo do último boot do equipamento. Quando o microcontrolador recebe a requisição SNMP de último boot, ele retorna uma variável que é incrementada a cada segundo.

Uma outra operação que é realizada no loop principal do microcontrolador é a atualização da informação mostrada no display LCD. A função que faz a modificação da informação no LCD recebe uma flag que representa qual index do menu o LCD deve prosseguir. Como padrão, existem 9 telas de menus. Esta função de atualização de LCD só é chamada quando uma interrupção é ativa no pino digital. Esta interrupção é um botão do tipo *pushbutton* e está localizada na parte frontal do equipamento.

Entretendo, alguns gigaPOPs possuem um display LCD mais robusto formado por uma matriz de 4 linhas onde são mostradas mais informações ao operador.

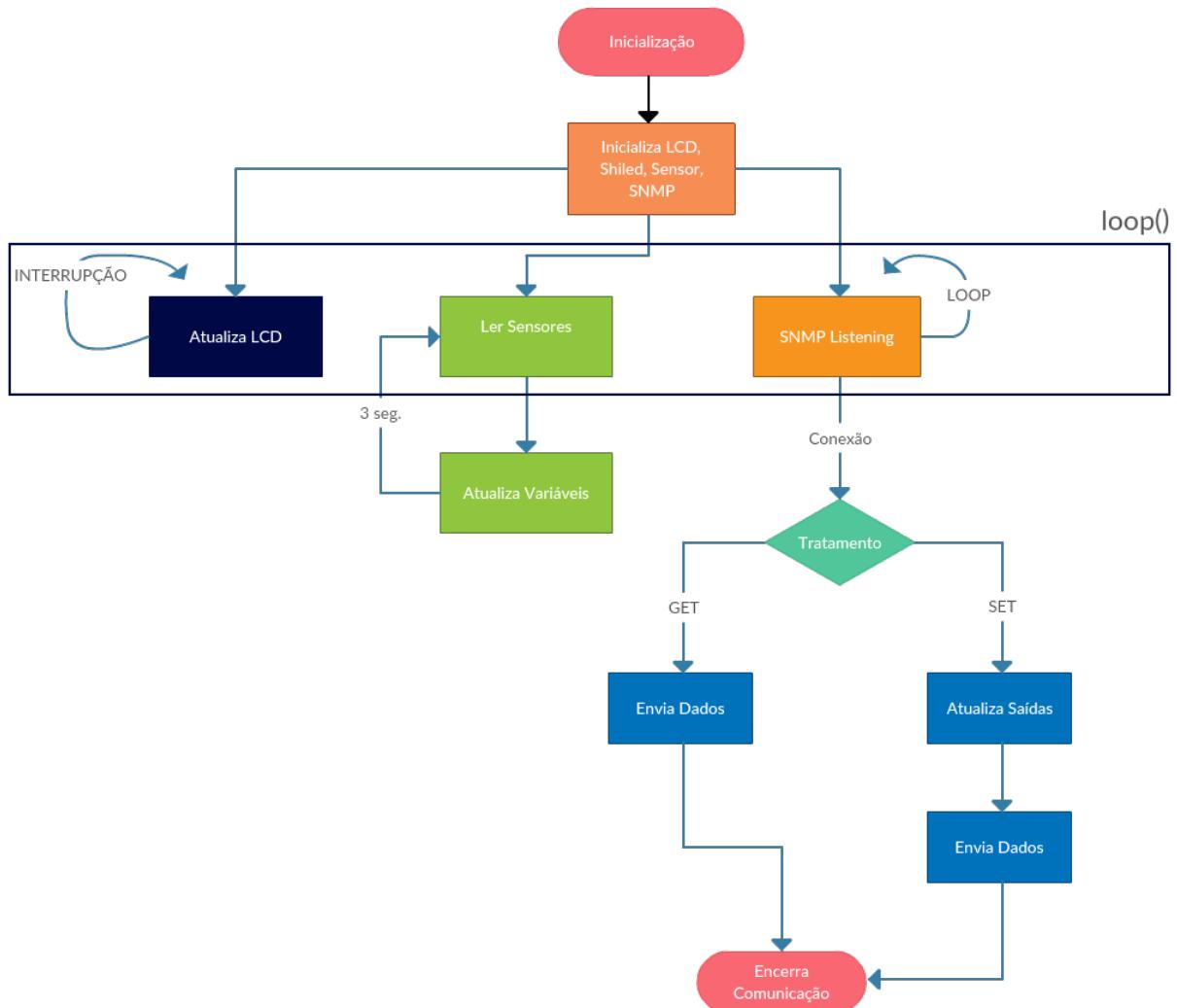


Figura 5.2: Fluxograma de Funcionamento do Sistema Embarcado

São mostradas as seguintes informações no display LCD:

- Endereço MAC
- Endereço IP
- Porta de comunicação SNMP
- Gateway padrão
- Máscara de rede
- Estado da porta do rack
- Estado do ar-condicionado
- Nome da estação
- Último boot
- Temperatura dos sensores
- Umidade
- Versão do firmware

A terceira e última operação é a conexão SNMP. Uma vez inicializada, o microcontrolador fica aguardando conexões na porta especificada. Quando uma conexão é estabelecida, uma função de decodificação é chamada. Nesta função, o OID do SNMP é comparado com todos os OIDs cadastrados. Após identificar qual operação realizar, o microcontrolador encerra a conexão SNMP e executa a operação. Caso seja uma operação de consulta, uma String contendo o dado solicitado é retornado antes do encerramento da conexão.

O código responsável pela comunicação do agente SNMP foi baseado na biblioteca Agentuino. Esta biblioteca foi desenvolvida por Eric C. Gonet em 2010 e possui licença de código aberto distribuída gratuitamente. O desenvolvimento da biblioteca foi interrompido e o estágio da versão atual é alfa. Consequentemente, a biblioteca não é estável e não funciona em paralelo com outras instruções no microcontrolador. Além disso, esta biblioteca não é recomendada para uso no microcontrolador ATmega328p devido a limitações físicas de memória SRAM e Flash (Gonet, 2010).

Foram necessários muitas modificações no código fonte da biblioteca para a adequação ao sistema proposto. A primeira alteração que foi realizada foi a simplificação dos comandos SNMP para apenas GetRequest. Não é possível usar outros comandos de gerenciamento, como o Trap ou SetRequest. Outras alterações na estrutura do código também foram realizadas.

Além da biblioteca Agentuino, foram usadas as bibliotecas Streaming LibraryFlash Library e MemoryFree Library desenvolvida por Hart (2013) e são executadas em conjunto com a Agentuino.

5.2 Pinagem

O projeto do sistema de hardware foi realizado usando o software ¹Eagle PCB Design. A figura 5.3 mostra a ligação interna do microcontrolador com a placa eletrônica. O microcontrolador foi conectado a uma conector de 14 vias que é conectado diretamente da placa eletrônica.

¹ Eagle PCB é um software de design de circuitos eletrônicos com criação de esquemático e layout de placa de circuito impresso. A licença é gratuita para estudantes e para uso acadêmico.

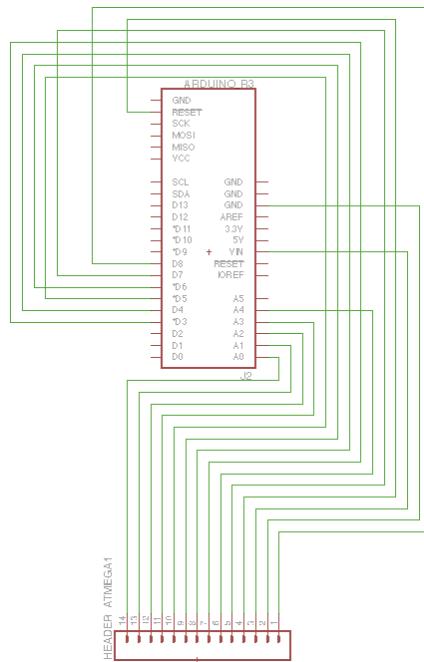


Figura 5.3: Ligação do Microcontrolador

A pinagem do microcontrolador com os dispositivos de entrada e saída é mostrada na tabela 5.1. Devido a limitação de quantidade de portas digitais, foram usadas as portas analógicas como portas digitais para controlar o display LCD.

Tabela 5.1: Pinagem do Microcontrolador

Pino Microcontrolador	Função
PIN A0	Sensor Termistor Analógico
PIN A1	LCD D7 Digital
PIN A2	LCD D6 Digital
PIN A3	LCD D5 Digital
PIN A4	LCD D4 Digital
PIN 3	Relé Ar-Condicionado
PIN 4	Relé Porta do Rack
PIN 5	Botão do Menu
PIN 7	Sensor DHT Digital
PIN 8	LCD Enable
PIN 9	LCD RS

Capítulo 6

6 Circuito Eletrônico

O circuito eletrônico do dispositivo desenvolvido é composto por três tipos de circuitos acoplados em uma placa de circuito impresso. O primeiro circuito é uma fonte linear de alimentação que converte a tensão da rede elétrica de 220/110V AC para uma tensão de 12V DC. O segundo circuito é uma fonte de alimentação chaveada que converte 12V para o nível de operação TTL do circuito, ou seja, 5V. Finalmente temos o circuito responsável pelo controle do sistema e conexão entre microcontrolador e dispositivos de entrada e saída.

6.1 Circuito de Alimentação

6.1.1 Fonte Principal de 12V

Uma fonte linear de tensão foi desenvolvida para alimentar todo o circuito. O dimensionamento dos parâmetros da fonte foram definidos de acordo com os dispositivos eletrônicos usados. O componente de maior potência do sistema é a fechadura magnética, que tem um consumo de corrente de aproximadamente 400 mA em 12V DC. Já para o microcontrolador, foi necessário realizar um cálculo teórico de consumo a partir das especificações do fabricante. O resultado obtido foi de aproximadamente 200 mA de consumo de corrente em 5V DC.

A partir desses dados, foi projetado uma fonte linear de tensão de 1A de corrente de saída para 12V DC. O transformador usado foi um de derivação central com entrada 110V ou 220V selecionados através de um chaveamento entre entradas no primário do transformador. A figura 6.1 mostra o esquemático do circuito projetado.

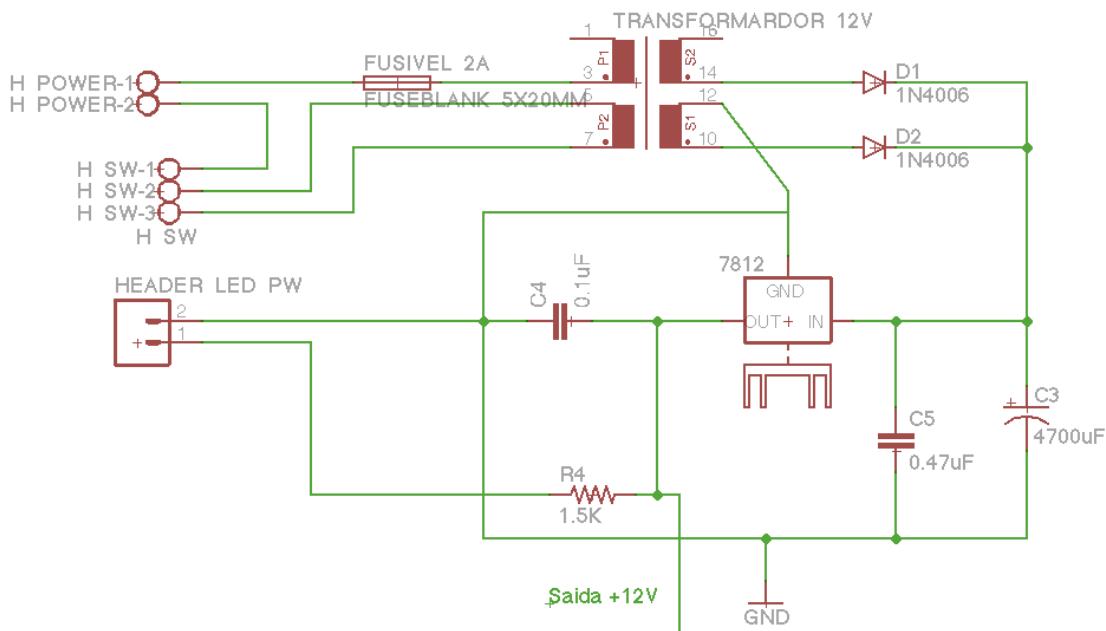


Figura 6.1: Circuito Eletrônico Fonte Linear

Foi adicionado um fusível de proteção na entrada do primário do transformador de 2A para evitar eventuais acidentes de sobrecarga do sistema. O cálculo de corrente máxima foi obtido mensurando-se a corrente de consumo total do circuito, que foi de aproximadamente 1A em 220V AC. A saída do secundário do transformador passa por um circuito retificador de onda completa composto por dois diodos de corrente máxima de 1A cada. A saída do retificador é então conectada a um regulador de tensão de uso geral LM7842. Um LED que indica o funcionamento da fonte foi acoplado ao circuito usando um resistor limitador de corrente. Para mudar a tensão de entrada entre 110V ou 220V, foi acionado uma chave do tipo seletora que se conecta a placa eletrônica através de um conector do tipo borne. Devido a alta potência de energia dissipada na forma de calor pelo regulador de tensão, foi necessária acoplar-lo a um dissipador de calor de alumínio e adicionar um fan ao equipamento.

6.1.2 Fonte Secundária de 5V

O microcontrolador, assim como os sensores, Shield Ethernet e display LCD funcionam com alimentação de 5V. Para que o sistema obtivesse uma alta eficiência energética, foi projetada um conversor de tensão do tipo *step-down* (*buck*) que converte os 12V da fonte principal em 5V para alimentação do circuito. Um conversor buck é usado quando precisamos diminuir a tensão DC de um circuito. O circuito consiste em um transistor que faz o chaveamento da tensão na saída e um circuito de *flywheel* composto por um indutor, um capacitor e um diodo.

A eficiência de fontes de tensão buck são muito superiores a de fontes lineares baseadas em transformadores e reguladores de tensão como mostrado por Dantas et al. (2016). A figura 6.2 mostra o circuito eletrônico do conversor desenvolvido.

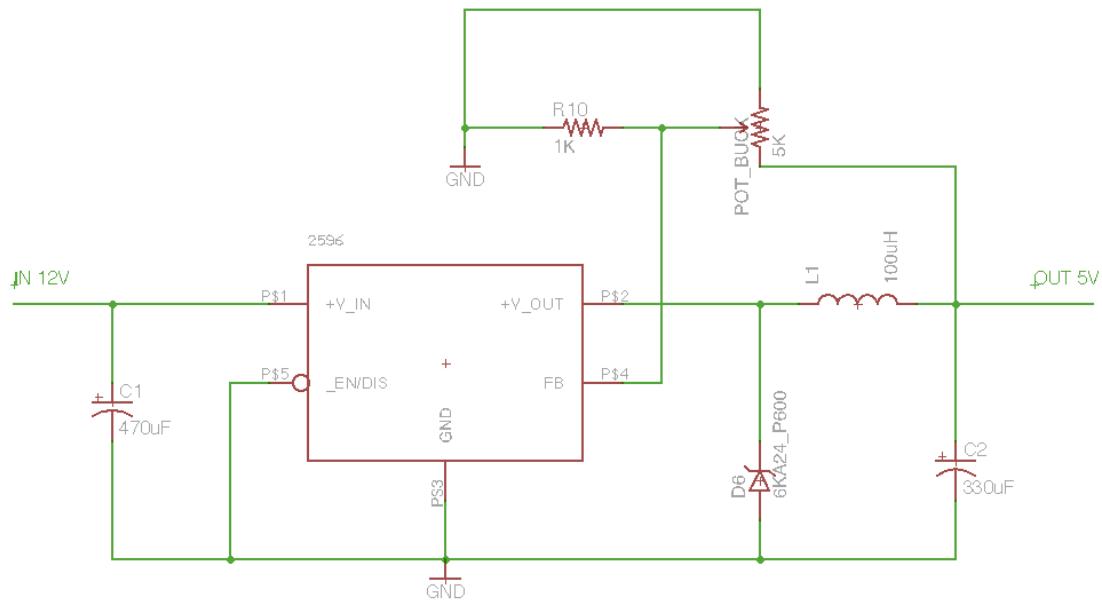


Figura 6.2: Conversor de Tensão Buck

O circuito integrado utilizado foi o LM2596 no modo ajustável de tensão. Para cálculo dos componentes necessários foi usado as especificações passadas pelo fabricante. Este CI é capaz de suprir até 3A de corrente em 5V. A frequência de chaveamento é de 150 kHz. Para se obter uma tensão de 5V, foi usando um sistema de feedback que é composto por um divisor de tensão entre dois resistores R_1 e R_2 .

O cálculo dos valores destes resistores é apresentado na equação 6.1 que foi disponibilizada pelo fabricante Texas Instruments (Texas Instruments Incorporated, 2016).

$$V_{\text{out}} = V_{\text{REF}} \left(1 + \frac{R_2}{R_1} \right) \quad (6.1)$$

$$V_{\text{REF}} = 1,23 \text{ V} \quad (6.2)$$

A partir das equação 6.1 e 6.2, e escolhendo o valor de $R_2 = 1 \text{ k}\Omega$ podemos simplificar o cálculo na equação 6.3.

$$R_2 = R_1 \left(\frac{V_{\text{OUT}}}{V_{\text{REF}}} - 1 \right) \quad (6.3)$$

Portanto, obtemos o valor de $R_2 = 3,06 \text{ k}\Omega$. O indutor usado foi um de núcleo de ferrite de valor 100 mH. A escolha de um indutor com alta indutância seu deu para que variações bruscas de consumo de corrente, como o fechamento da porta do rack por exemplo, tenham pouca influencia na alimentação do circuito de controle. Em outras palavras, buscou-se reduzir o *ripple* da saída do conversor de tensão.

6.2 Circuito de Controle

O circuito de controle é responsável pelo o acionamento do ar-condicionado e por controlar os sensores externos de temperatura. Este circuito recebe a alimentação das duas fontes de tensão e faz a conexão entre o microcontrolador, o display LCD, os LEDs, o botão do menu, a fechadura magnética e o relé de estado sólido que aciona o ar-condicionado.

Foram usados dois relés para o acionamento da fechadura magnética e do relé de estado sólido do acionamento do ar-condicionado. Um diodo de *flyback* foi conectado em paralelo a bobina de cada relé para eliminar o alto potencial negativo gerado pela bobina magnética quando o relé é desenergizado. Isso se dá ao fato de que indutores se opõe a varrições bruscas de corrente, ou seja, quando desligamos o indutor da fonte de alimentação, a energia armazenada nele tende a se manter constante e o diodo de *flyback* é usado para dissipar essa energia e evitar que ela cause danos ao circuito. O acionamento dos relés é feito por resistores de uso geral de polarização NPN para pequenas cargas. Em cada transistor foi conectado um resistor de 1 $\text{k}\Omega$ em sua base. Como os dois relés estão realizando o chaveamento de uma tensão de 12V, foram conectados resistores de 1,5 $\text{k}\Omega$ para adicionar um LED de indicação em cada um deles. Um relé de estado sólido de 40A foi usado para acionar o ar-condicionado. A figura 6.3 mostra o circuito de controle desenvolvido.

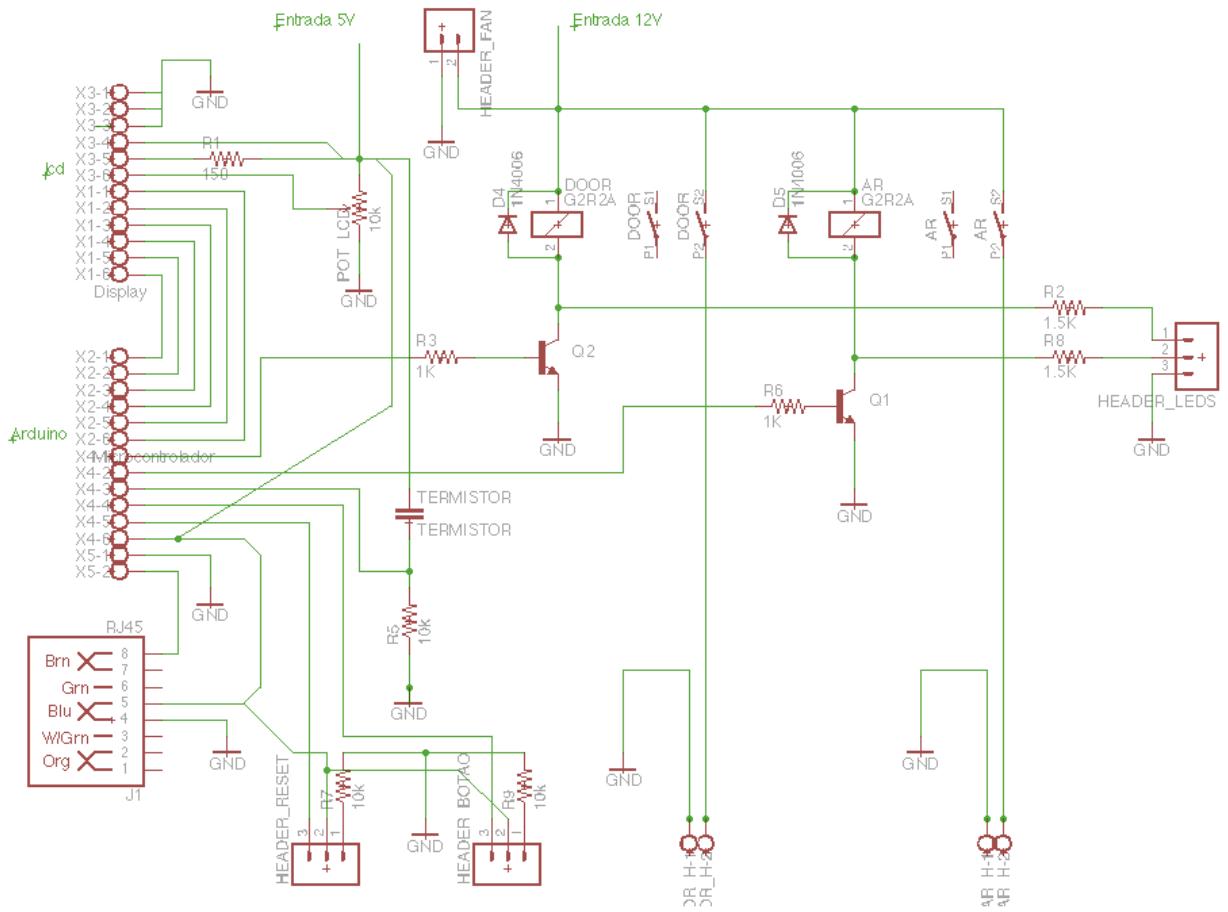


Figura 6.3: Circuito de Controle

6.3 Placa de Circuito Impresso

A placa de circuito impresso também foi projetada usando o software Eagle PCB. A placa contém todos os circuitos e possui dimensões de 20x16 cm. O projeto foi feito em uma camada de cobre simples, e foi necessário usar conectores do tipo jumpers de cobre na parte superior da placa. A figura 6.4 mostra a parte superior da PCB. Nela foi impressa a legenda dos componentes e as trilhas de cobre feitas por jumper.

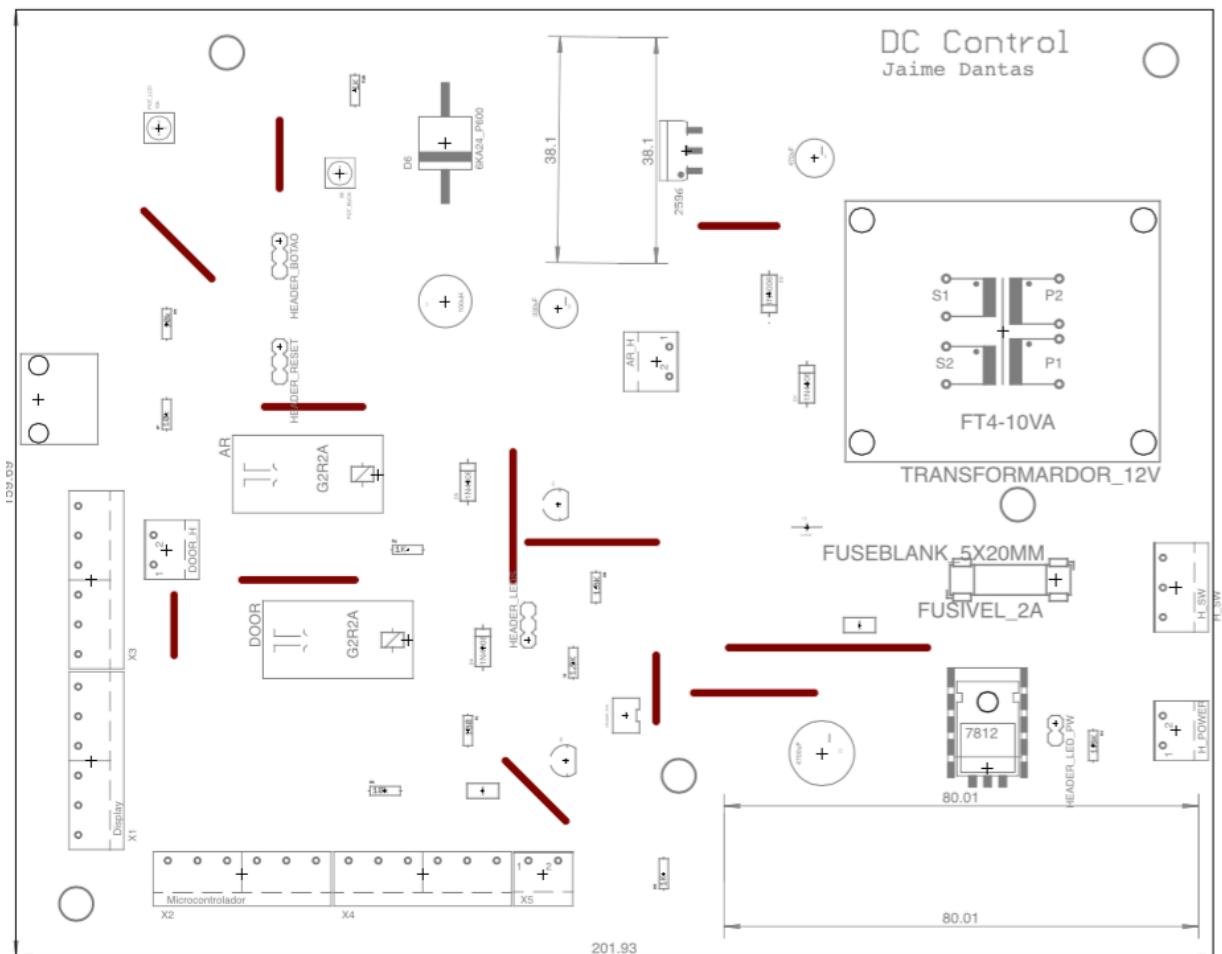


Figura 6.4: Parte Frontal da PCB

A figura 6.5 mostra a parte inferior da PCB que contém todas as trilhas de cobre do circuito. O processo de fabricação de PCB usado foi do tipo de corrosão por perclorato de ferro. A imagem inferior da PCB foi impressa em um papel do tipo transfer usando uma impressora a laser e toner de cor preta. Com o auxílio de um ferro de passar roupas, a imagem impressa foi transferida para a placa de cobre e esta então foi inserida no perclorato de ferro para corrosão.

Após a definição de todas as trilhas de cobre na placa, os furos para inserção dos componentes eletrônicos foi realizado usando uma furadeira e brocas de acordo com o diâmetro de cada componente.

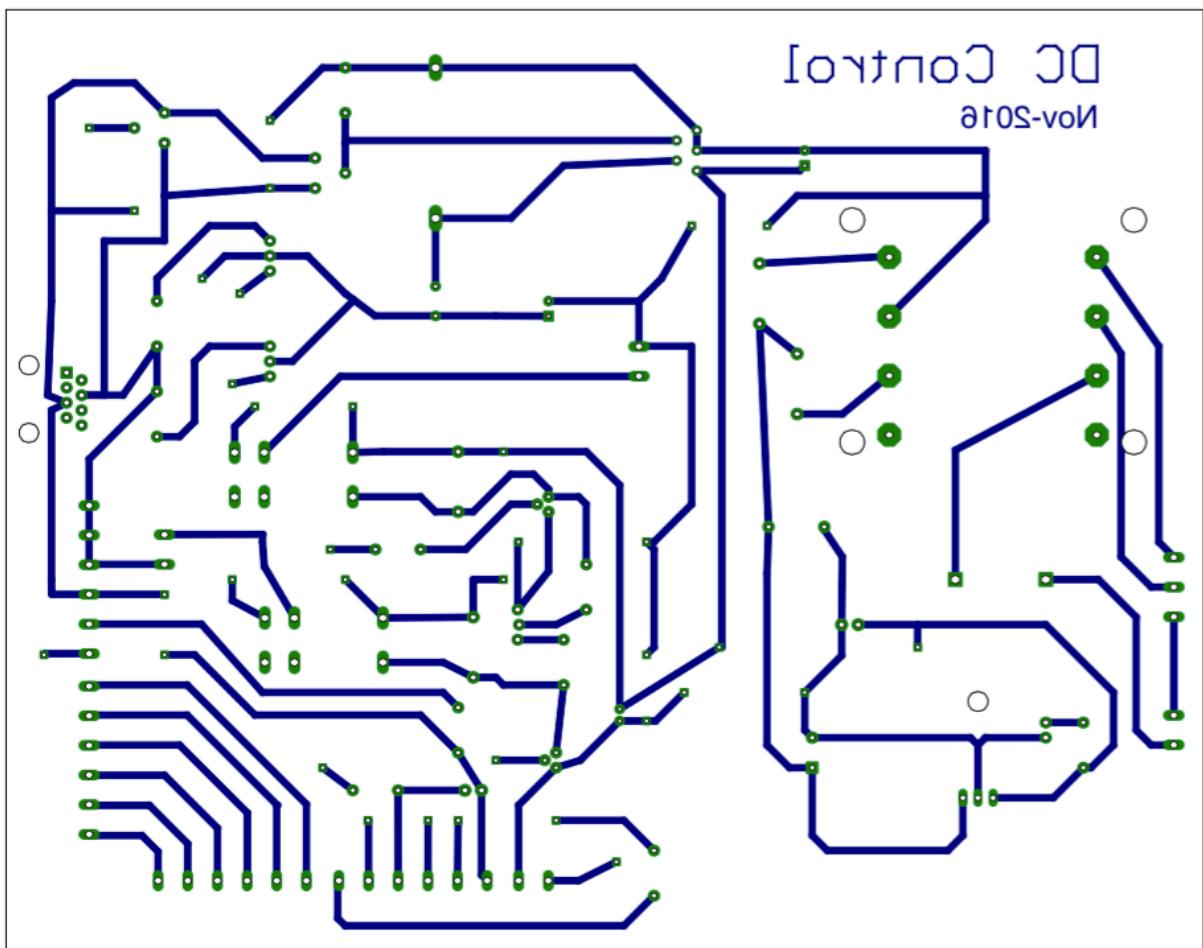


Figura 6.5: Parte Inferior da PCB

Capítulo 7

7 Sistema de Controle e Monitoramento Java

O sistema de gerenciamento, controle e monitoramento de dados foi desenvolvido usando a linguagem Java através da ferramenta de desenvolvimento NetBeans IDE e foi chamado de DC Control. Foram utilizadas algumas bibliotecas Java para facilitar o desenvolvimento do programa assim como aumentar a confiabilidade do mesmo. O sistema criado faz a aquisição de dados e o envio de comandos para as estações usando unicamente o protocolo de gerenciamento SNMP. O sistema não possui banco de dados, portanto não é capaz de armazenar o histórico dos dados das estações.

Todo armazenamento e análise de dados é feito por um servidor de dados dedicado que executa uma aplicação de aquisição de dados de gerenciamento de redes pelo protocolo SNMP. Nos nossos testes, foi usado como software padrão o Zabbix. O sistema DC Control é compatível com o sistema operacional Linux, Windows e MacOS.

7.1 Linguagem de Programação Java

Java é uma linguagem de programação de alto nível que é usado em diversas aplicações de uso geral. A linguagem Java foi desenvolvida pela Sun Microsystems por um pequeno time de engenheiros conhecidos como "The Green Team" em 1991. Em 2009 a Oracle Corporation, tornando-se proprietária da linguagem Java (Beal, 2016).

Atualmente, a linguagem Java é amplamente usada em diversos dispositivos e plataformas. Assim como a linguagem C++, Java é uma linguagem orientada a objetos com características que a simplificam e que tornam a programação mais simplificada comparada com C++. Além disso, existem diversas plataformas de desenvolvimentos que dão suporte ao Java.

7.2 NetBeans

NetBeans é um software de desenvolvimento feito inteiramente em Java. A plataforma NetBeans permite o desenvolvimento de aplicações modulares não somente em Java mas também em outras linguagem como C/C++, PHP e HTML5. O NetBeans é compatível com o sistema operacional Microsoft Windows, Mac OS X, Linux, Solaris e

outras plataformas que tem suporte a máquina virtual Java (Wikipedia, 2016). O sistema foi desenvolvido com a versão 8.0.2 do NetBeans no sistema operacional MacOS X Yosemite 10.10.5.

7.3 Bibliotecas

7.3.1 API SNMP para Java SNMP4J

O SNMP4J é uma biblioteca Java de licença gratuita de código aberto. Ela implementa todas as funcionalidades do protocolo SNMP para versões do JavaTM SE 1.4 ou superior. A biblioteca suporta gerentes e agentes SNMP e foi desenvolvida com base no SNMP++, que é muito conhecido na desenvolvimento de aplicações SNMP para C++. Além da versão gratuita, a biblioteca possui suporte diferenciado para a versão comercial (SNMP4J, 2015).

7.3.2 API Gráfica para Java JfreeChart

JfreeChart é uma biblioteca gráfica para Java de código aberto distribuída de forma gratuita pela licença *GNU Lesser General Public Licence (LGPL)*. Esta biblioteca é a mais usada para criação de gráficos e possui uma vasta documentação. Além disso, JfreeChart oferece suporte a Swing do Java e JavaFX, sendo possível trabalhar com vetorização de componentes.

A biblioteca foi desenvolvida em 2000 por David Gilbert, e é hoje a maior e mais usada entre os programadores Java (JFreeChart, 2016).

7.4 Especificações do Sistema

O sistema foi modelado de acordo com as necessidades da empresa PoP-RN e baseado nos sistemas existentes no mercado NetBotz¹ do fabricante APC e no SNMP Environmental Sensors da fabricante ITWatchDogs². Entretanto, os sistemas citados acima também possuem uma interface de gerenciamento WEB, além de oferecerem

¹ NetBotz é uma solução de hardware e software desenvolvida pela APC da Shnelder Electric que possui gerenciamento SNMP e suporte a diversos sensores e câmeras de vídeo e tem um custo de 1.055,00 USD (Schneider Electric, 2016).

² ITWatchDogs é uma fabricante americana de equipamentos de monitoramento de data centers através do protocolo SNMP e sensoriamento. O modelo Watchdog 1250 possui suporte a sensores e gerenciamento remoto e tem um custo de 579,00 USD (ITWatchDogs, 2016).

mais funcionalidades dos sistema desenvolvido DC Control. O PoP-RN possui 10 gigaPOPs localizados em todo estado do Rio Grande do Norte. Cada gigaPOP possui pelo menos um rack de comunicação, um ar-condicionado, um nobreak, e diversos ativos de rede de fibra óptica.

Na maioria dos gigaPOPs, existem mais de um ar-condicionado do tipo Split, e é feito a alternância manual dos mesmos para prolongar a vida útil do conjunto. Um dos principais requisitos do sistema desenvolvido era acionar os acondicionados de forma remotamente e também fazer o rodízio de forma automática dos mesmos. Além disso, caso um ar-condicionado apresente falha, o sistema deveria acionar o ar-condicionado auxiliar de forma automática e comunicar para o operador.

Além disso, a abertura e fechamento da porta do rack de comunicação deveria ser feita por um operador remotamente. Finalmente, o proposito principal do sistema desenvolvido era monitorar a temperatura, humidade e outros sensores usando o protocolo de gerenciamento de redes SNMP.

7.5 Diagramas UML

A Linguagem de Modelagem Unificada (UML) é uma linguagem de uso geral para a modelagem de sistemas computacionais que fornecem uma padronização na visualização do design do sistema.

7.5.1 Diagrama de Caso de Uso

O diagrama de caso de uso apresentado na figura 7.1 descreve as funcionalidades do sistema DC Control. O diagrama faz um levantamento geral de todas as opções oferecidas pelo sistema.

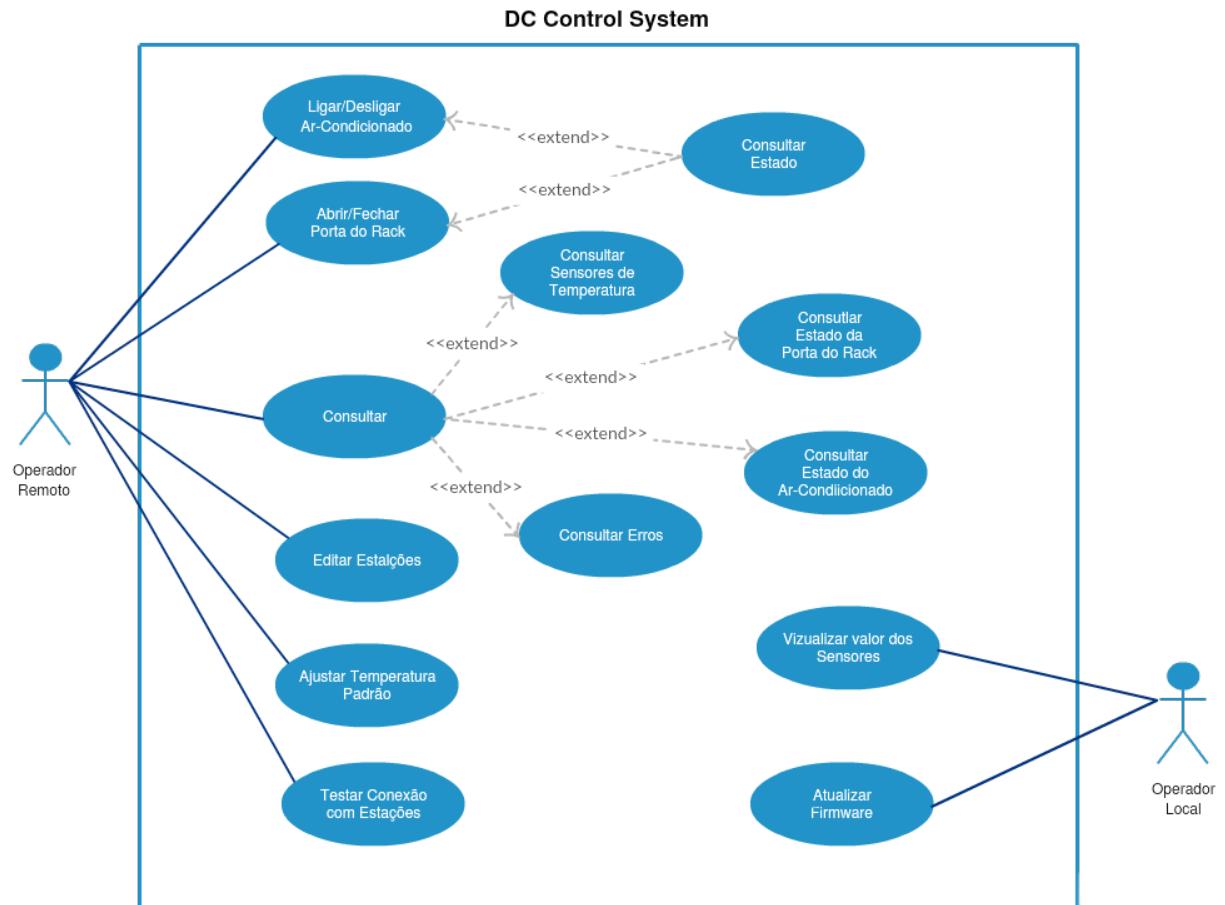


Figura 7.1: Diagrama de Uso Sistema DC Control

7.5.2 Diagrama de Atividades

Diagrama de atividades mostram um gráfico em fluxo do controle de uma atividade em específico do sistema. A figura 7.2 mostra as interações que ocorrem durante o processo de abertura da porta do rack.

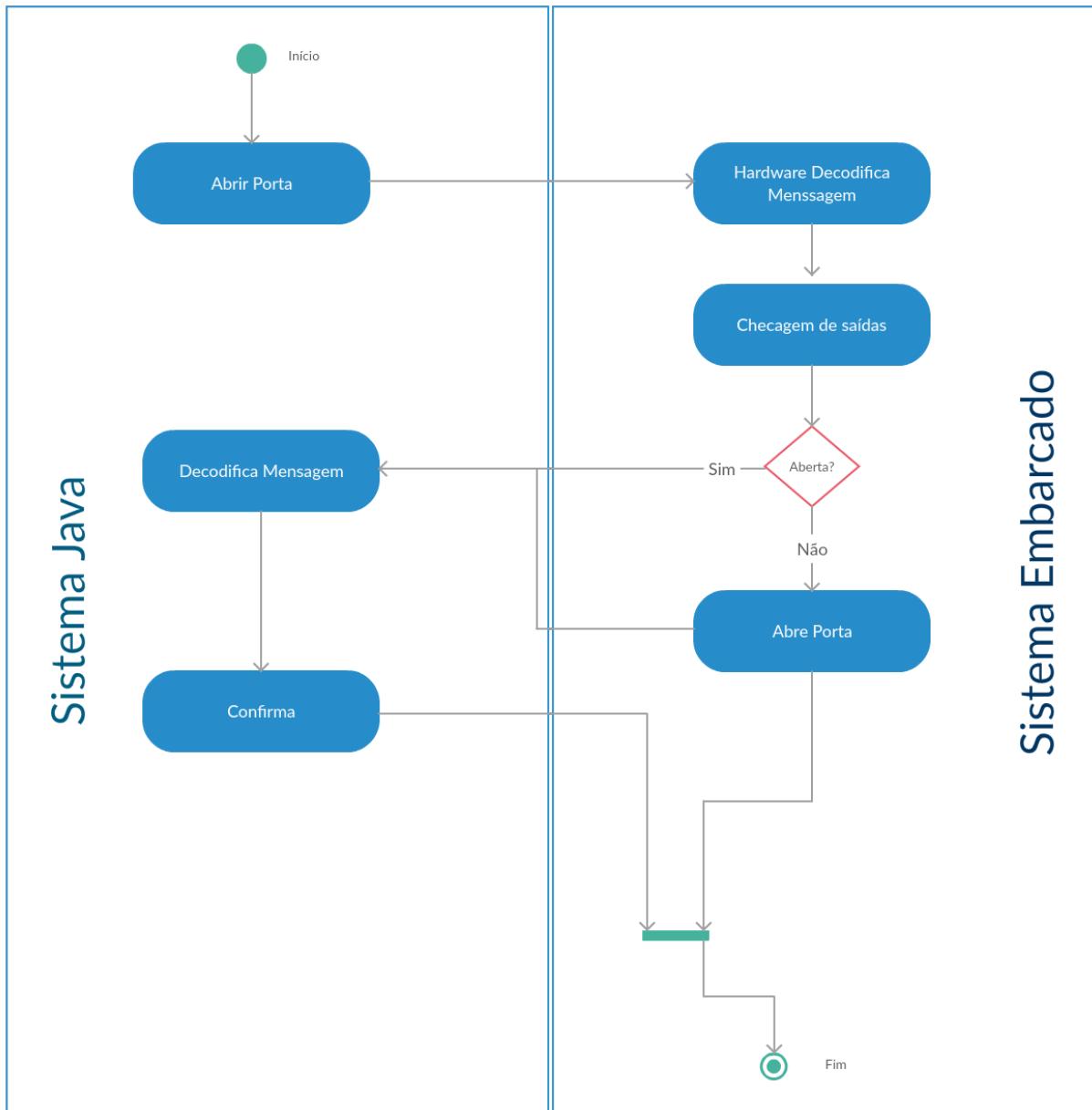


Figura 7.2: Diagrama de Atividade de abertura de porta

O sistema DC Control faz uma checagem para verificar se a porta do rack foi aberta com sucesso. O mesmo processo também ocorre com o acionamento remoto do ar-condicionado. A figura 7.3 mostra como ocorre o acionamento automático do ar-condicionado quando a temperatura do ambiente ultrapassa o valor máximo previamente configurado (sp).

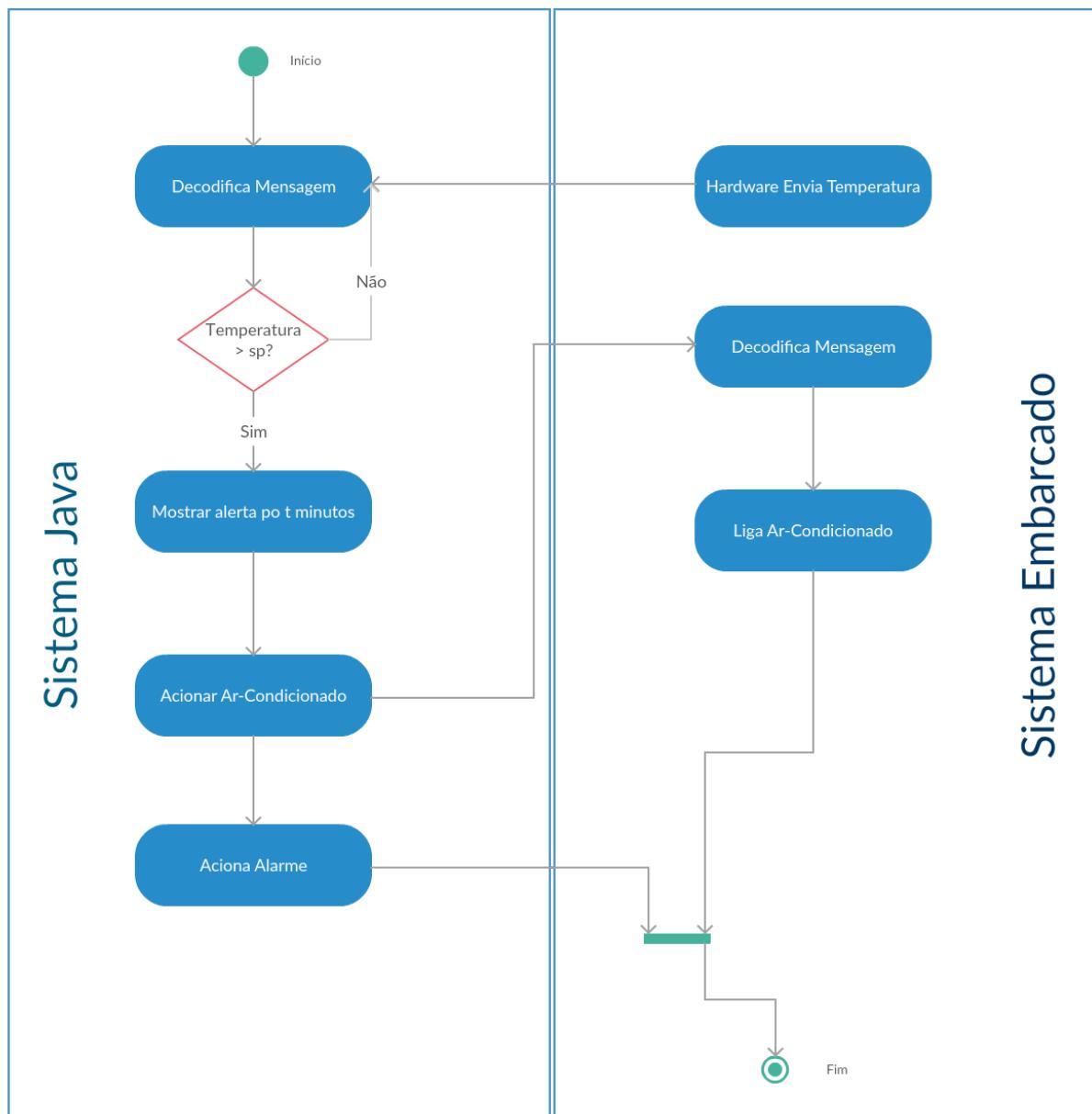


Figura 7.3: Diagrama de Atividade de Acionamento do Ar-Condicionado

7.6 Funcionamento do Sistema

O sistema DC Control possui muitas funções tanto de gerenciamento quanto de controle. Os módulos do sistema estão divididos em três: Monitoramento, Controle e Gerenciamento. A seguir são discutidos as características e funcionalidades de cada módulo em específico.

7.6.1 Módulo de Monitoramento

O sistema DC Control tem como principal característica monitorar a temperatura das estações cadastradas. Como não foi usado um banco de dados para armazenar

as informações coletadas pelo DC Control, o armazenamento dos dados se dá de forma estática. Dessa forma, toda vez que o operador iniciar o sistema ele iniciará a aquisição de dados e finalizará quando o programa for fechado.

O monitoramento das estações será feito através de um hardware específico composto por um Raspberry Pi B conectado a uma TV de 42''. A tela de monitoramento foi ajustada ao padrão da TV, com resolução de Full HD.

Assim que o usuário abre o sistema, é necessário realizar uma autenticação para usar o sistema. Foram cadastrados usuários com todas as permissões e usuários para monitoramento. A figura 7.4 mostra a tela de login do sistema DC Control



Figura 7.4: Tela de Login

Todos os usuários e senhas são gravados no próprio código do programa em uma classe específica para usuários. Portanto, para mudar a senha de um usuário, é necessário editar o código na classe e compilar o programa novamente para gerar o executável.

Após realizar o login no sistema, o usuário terá que escolher quais estações ele deseja realizar o monitoramento. O sistema DC Control tem a capacidade de realizar o gerenciamento simultâneo de até 10 estações. A figura 7.5 mostra a tela onde o usuário marca quais estações estão ativas para o monitoramento.

A aquisição de dados simultâneo para as diferentes estações é realizada com o uso de threads. Cada estação tem uma thread que realiza a requisição SNMP de forma independente das outras threads do sistema. A figura 7.6 mostra todas as threads do sistema. A geração dos gráficos, assim como a atualização dos mesmos é feitos por threads independentes.

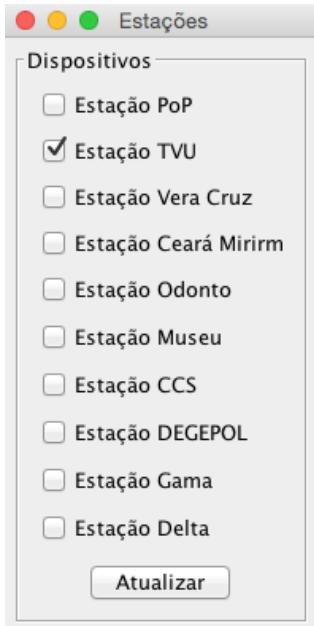


Figura 7.5: Tela de Escolha das Estações Ativas

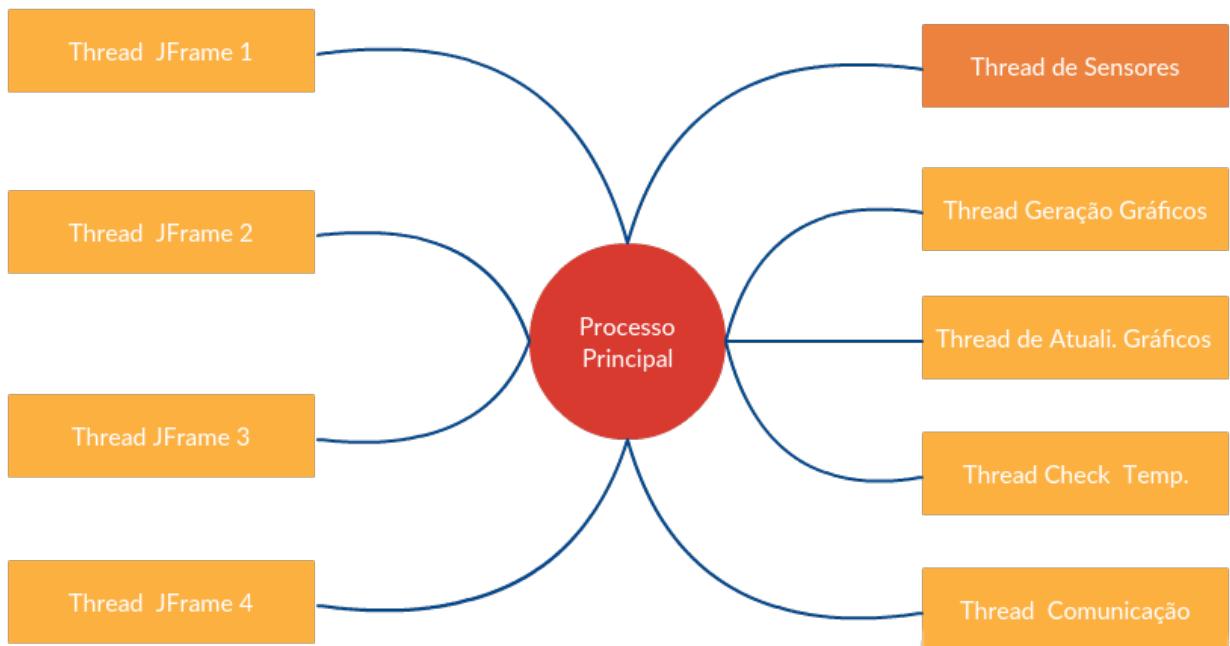


Figura 7.6: Threads do Sistema

As threads representadas por "JFrame" são responsáveis por modificar componentes dinâmicos das janelas do programa como barra de progresso e tarefas independentes. Também existem uma thread para verificar se a temperatura da estação excedeu a temperatura configurada pelo usuário. E por fim, a thread de comunicação verifica se há erros de comunicação com as estações.

Após realizar o login e escolher as estações ativas, a tela inicial de monitoramento é mostrada na figura 7.7. Essa é a tela que é mostrada na TV, onde é feito o

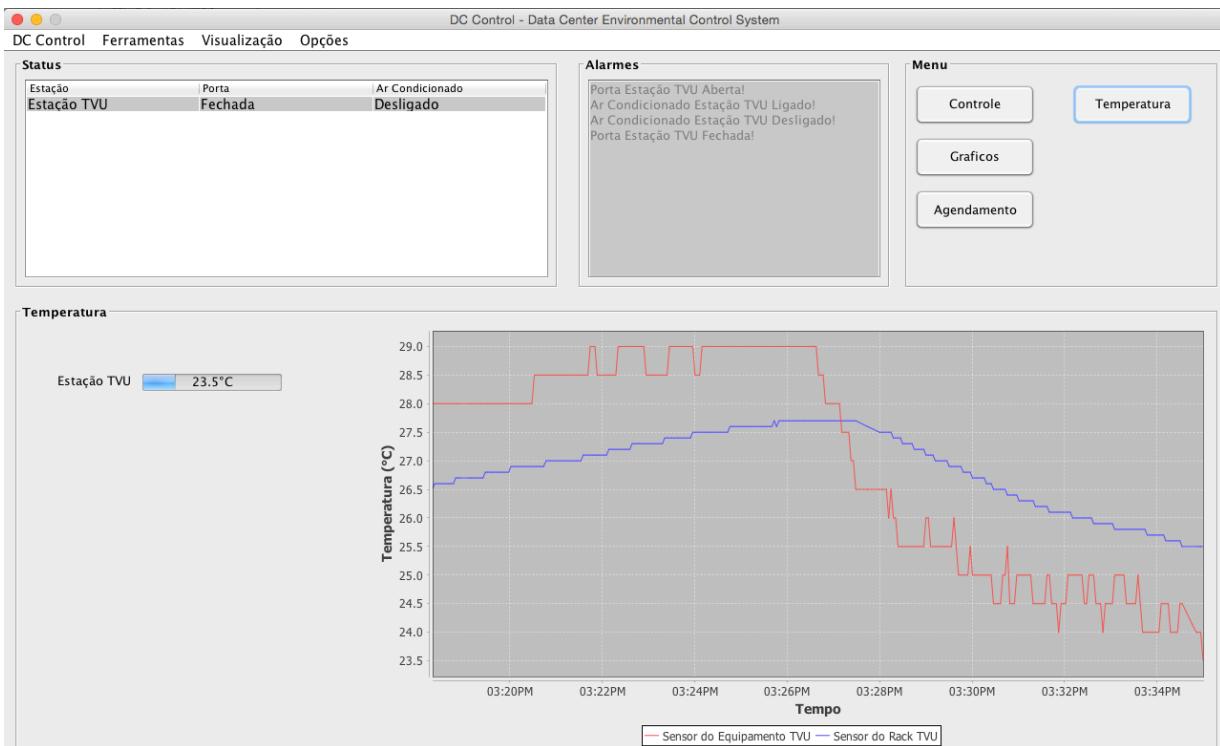


Figura 7.7: Threads do Sistema

monitoramento visual das estações. Podemos encontrar na tela inicial todas as informações necessárias para realizar o gerenciamento das estações.

No canto superior esquerdo da se encontra uma lista com todas as estações conectadas. O sistema DC Control tem a capacidade de realizar o gerenciamento simultâneo de até 10 estações. Nesta lista, temos o nome da estação, o estado do ar-condicionado, e o estado da porta do rack. Apenas as estações que o usuário escolhe no login são mostradas na lista.

Já no canto superior central se encontra a uma tela com todos os alarmes do sistema. Esses alarmes são gerados quando gerados as seguintes ocorrências

- Porta do rack aberta
- Porta do rack fechada
- Ar-condicionado ligado
- Ar-condicionado desligado
- Temperatura acima do normal
- Erro de comunicação com a estação

No canto superior direito ficam localizados os botões para as principais funções do sistema. Na parte inferior esquerda são listadas todas as estações ativas e é mostrado a temperatura do rack de cada estação em uma barra. Os limites de temperaturas inferiores e superiores da barra são configurados de acordo com a temperatura média de cada gigaPOP.

Já na parte inferior direita temos o gráfico que mostra a temperatura de todas as estações ativas. Também é possível visualizar apenas a temperatura do rack ou apenas a temperatura interna no equipamento ajustando o parâmetro de visualização de gráficos, como mostrado na figura 7.8.

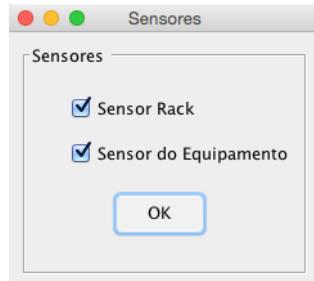


Figura 7.8: Tela de Seleção de Sensores

No gráfico é possível dar um zoom para visualizar a temperatura em um determinado tempo. O eixo do domínio x é o tempo atual da máquina que está executando o sistema. Caso deseje uma análise mais aprofundada, o operador pode abrir no menu principal a tela contendo todos os gráficos gerados pelo sistema. A figura 7.9 mostra a temperatura de todos sensores das estações. Nela é possível visualizar a humidade, estado da porta do rack e o estado do ar-condicionado, temperatura do rack e do equipamento de todas as estações.

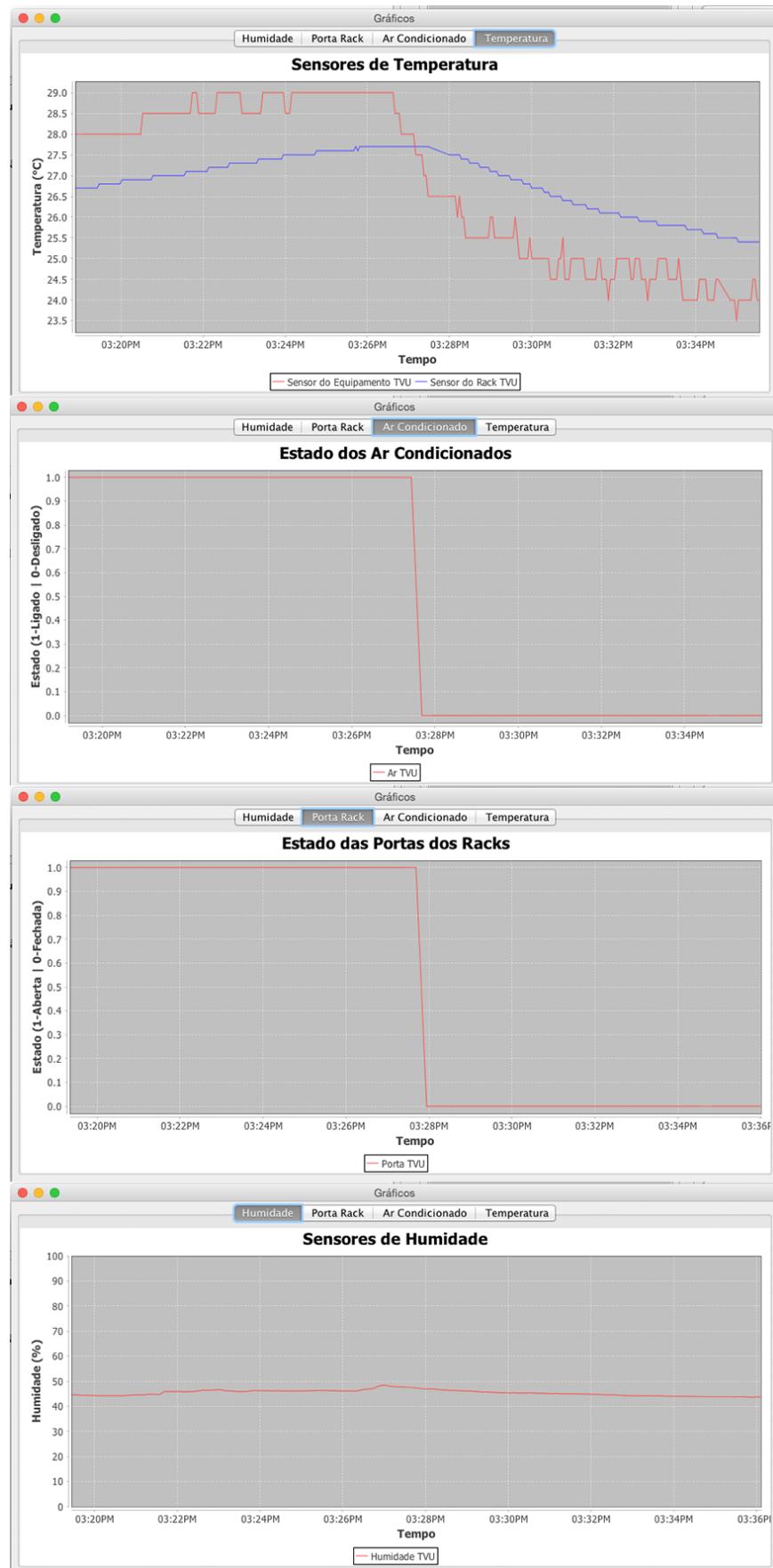


Figura 7.9: Tela dos Gráficos

7.6.2 Módulo de Controle

O sistema DC Control é capaz de controlar até 10 estações independentes. O operador pode controlar a abertura e fechamento da porta e o acionamento do ar-condicionado. Além disso, o sistema possui um algoritmo de controle para o acionamento do ar-condicionado caso ocorra uma falha no ar-condicionado principal ou a temperatura do ambiente aumente ao ponto de superar a temperatura máxima permitida configurada pelo operador.

A figura 7.10 mostra a tela de configuração da temperatura limite das estações. Podemos configurar não somente a temperatura mas também o tempo limite que o sistema deve esperar para que o alarme seja de fato gerado. Esse tempo limite é necessário para não gerar alarmes temporários quando a temperatura atinge a máxima por apenas poucos instantes.

Estação	Temperatura	Tempo
Estação PoP	40 <input type="button" value="▲"/> <input type="button" value="▼"/> C	30 <input type="button" value="▲"/> <input type="button" value="▼"/> seg.
Estação TVU	25 <input type="button" value="▲"/> <input type="button" value="▼"/> C	30 <input type="button" value="▲"/> <input type="button" value="▼"/> seg.
Estação Vera Cruz	40 <input type="button" value="▲"/> <input type="button" value="▼"/> C	30 <input type="button" value="▲"/> <input type="button" value="▼"/> seg.
Estação Ceará Mirim	40 <input type="button" value="▲"/> <input type="button" value="▼"/> C	30 <input type="button" value="▲"/> <input type="button" value="▼"/> seg.
Estação Odonto	40 <input type="button" value="▲"/> <input type="button" value="▼"/> C	30 <input type="button" value="▲"/> <input type="button" value="▼"/> seg.
Estação Museu	40 <input type="button" value="▲"/> <input type="button" value="▼"/> C	30 <input type="button" value="▲"/> <input type="button" value="▼"/> seg.
Estação CCS	40 <input type="button" value="▲"/> <input type="button" value="▼"/> C	30 <input type="button" value="▲"/> <input type="button" value="▼"/> seg.
Estação DEGEPOL	40 <input type="button" value="▲"/> <input type="button" value="▼"/> C	30 <input type="button" value="▲"/> <input type="button" value="▼"/> seg.
Estação Gama	40 <input type="button" value="▲"/> <input type="button" value="▼"/> C	30 <input type="button" value="▲"/> <input type="button" value="▼"/> seg.
Estação Delta	40 <input type="button" value="▲"/> <input type="button" value="▼"/> C	30 <input type="button" value="▲"/> <input type="button" value="▼"/> seg.

Figura 7.10: Tela de Configuração da Temperatura

Quando ocorrem manutenções nos gigaPOPs, a porta da sala onde os equipamentos se encontram é aberta, elevando a temperatura do ambiente a valores bem elevados. Para que o sistema não acione o ar-condicionado auxiliar, ele checa se a porta do rack está aberta. Caso a porta tenha sido aberta, o alarme não é gerado. Além disso, o operador pode desativar o controle automático de temperatura quando o gigaPOP está em manutenção.

O controle manual de abertura da porta do rack e acionamento do ar-condicionado é mostrado na figura 7.11. O operador seleciona qual estação ele deseja controlar, e o tipo do controle. Após enviar o comando para realizar a ação desejada, uma barra de progresso é ativa, e caso a operação ocorra com sucesso, um aviso é mostrado na tela assim como a imagem muda de acordo com o estado.

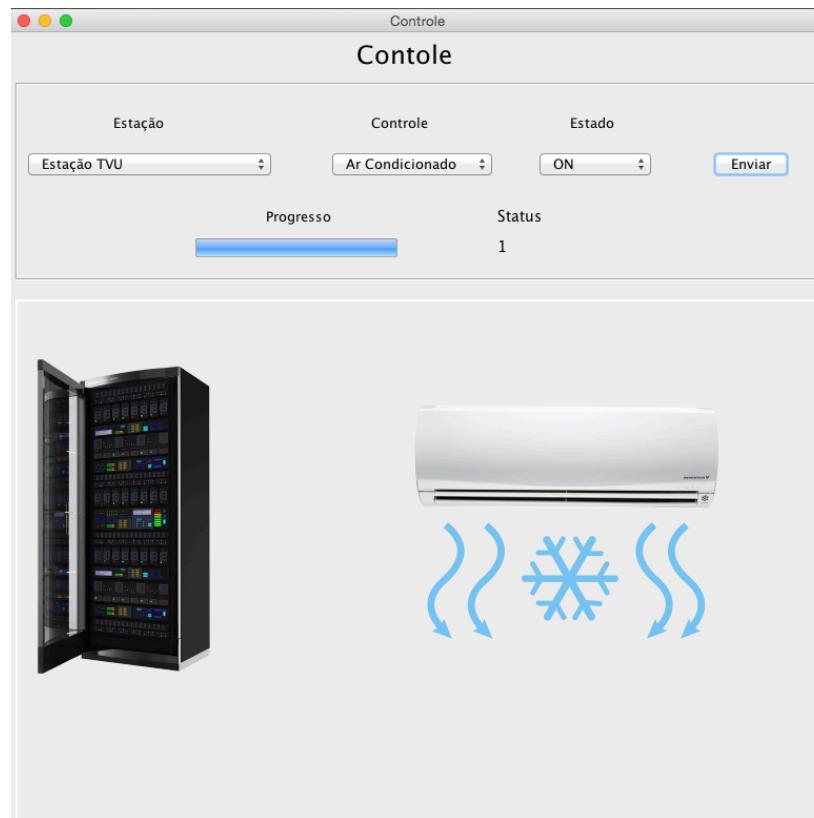


Figura 7.11: Tela de Controle da Estação

Como exemplo, foi desligado o ar-condicionado e fechada a porta do rack. A figura 7.12 mostra os avisos mostrados ao operador após realizadas estas ações.



Figura 7.12: Avisos de Ar-Condicionado Desligado e Porta Fechada

Quando a temperatura da sala atinge o limite máximo previamente configurado, um temporizador é disparado e uma tela informando o evento é mostrada para o operador. A figura 7.13 mostra uma barra de tempo para o acionamento do ar-condicionado. Caso a temperatura volte ao valores normais, o temporizador é zerado e a tela é fechada.

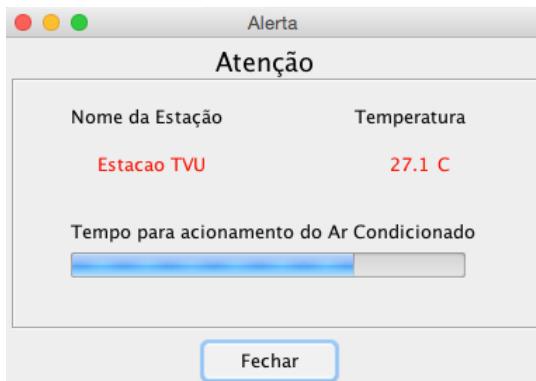


Figura 7.13: Tela de Temporizador do Acionamento do Ar-Condicionado

7.6.3 Módulo de Gerenciamento

O operador do sistema também dispõe de ferramentas de gerenciamento e diagnóstico das estações. É possível consultar todas as informações das estações cadastradas, assim como os dados dos servidores da banco de dados SNMP como mostra a figura 7.14.



Figura 7.14: Tela de Edição das Estações

O sistema obtém automaticamente o endereço IP do servidor de Controle, que é a máquina que o sistema está instalado. Além disso, também é possível realizar testes e diagnósticos de conexão entre o servidor (gerente) e as estações (agentes). A figura 7.15 mostra os testes que podemos realizar para diagnosticar falhas. O primeiro deles é o teste de resposta de *Ping*. Podemos configuras a quantidade de respostas que queremos, e ao final do teste o resultado será mostrado na *output*. Além disso, o sistema mostra a latência e a percentagem de perdas de pacotes do teste realizado. Uma barra de progresso mostra o andamento do teste.

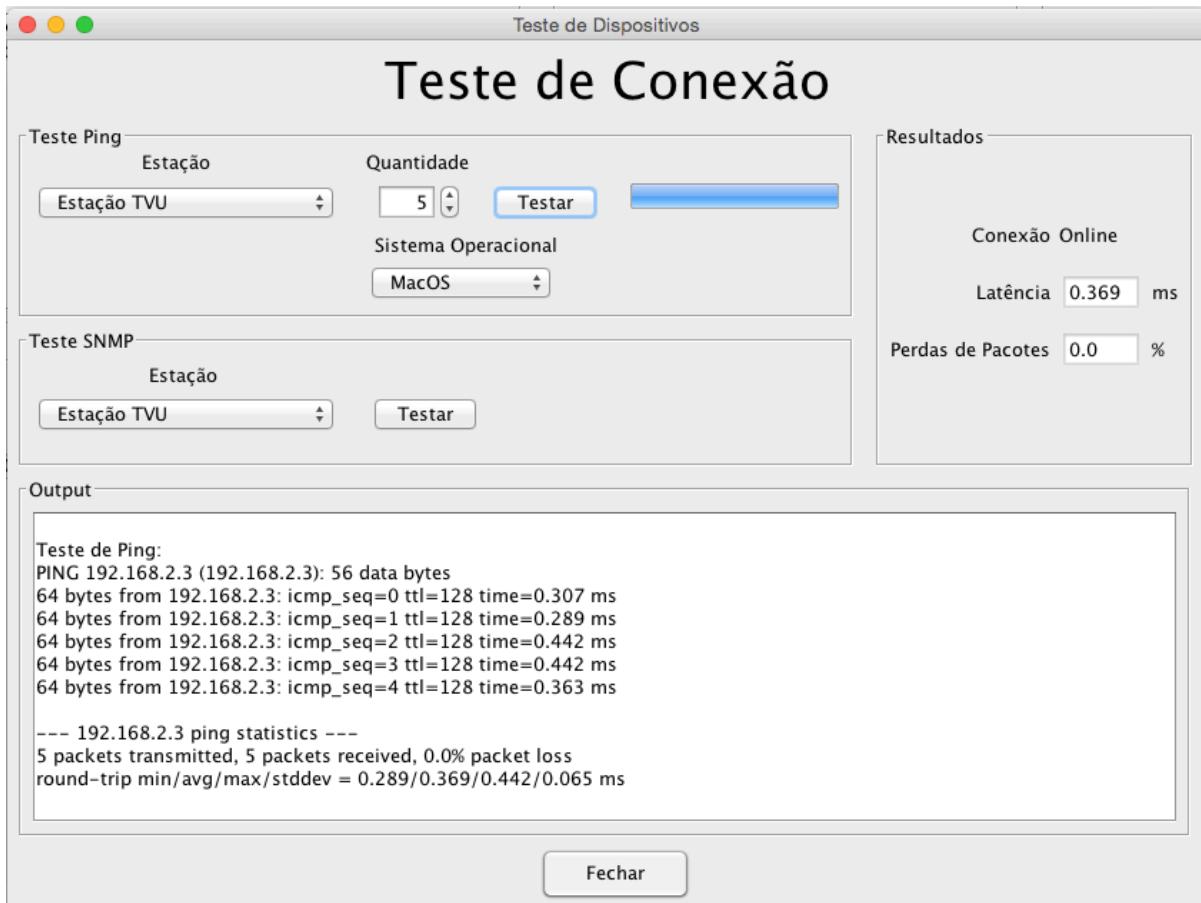


Figura 7.15: Tela de Diagnósticos de Falhas

Caso deseje fazer um teste de comunicação SNMP com o agente, o operador pode selecionar a estação para o teste SNMP é clicar em testar. Nesse teste, o sistema envia uma sequencia de requisições GetRequest para o agente e espera a resposta. Caso o teste ocorra com sucesso, o resultado é mostrado como na figura 7.16.

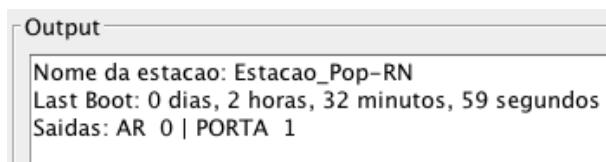


Figura 7.16: Retorno do Teste de Comunicação SNMP

Capítulo 8

8 Resultados Obtidos

Neste capítulo serão discutidos os resultados e testes que foram obtidos com o sistema proposto. Primeiramente será apresentado o sistema de hardware do equipamento. Na sequência, os resultados obtidos pelo software criado DC Control serão apresentados. Por fim, serão discutidos os resultados obtidos pelo monitoramento com o Zabbix.

8.1 Equipamento desenvolvido

O sistema microcontrolado projetado foi encapsulado em um equipamento de uso industrial para facilitar a instalação e o uso em ambientes de data centers. As dimensões foram projetadas de acordo com o padrão de equipamentos de redes para racks de comunicação. A figura 8.1 mostra o projeto do equipamento.



@JaimeDantas

Figura 8.1: Equipamento Projetado

O equipamento ocupa 2U de espaço em um típico rack de comunicação. Ele possui abas de fácil fixação no trilho do rack e a conexão do sensor de temperatura é

feita com uma cabo ethernet de uso geral, com conector RJ45. Todas essas soluções visam facilitar o instalação e manutenção do equipamento.

Na parte frontal se encontram o display LCD, que mostra informações úteis para o operador, LEDs que indicam o estado da porta do rack e do ar-condicionado, um botão de menu, uma porta USB padrão B para atualização de firmware, e uma porta Ethernet para conexão do equipamento a rede de comunicação. A figura 8.2 mostra o equipamento que foi desenvolvido.



Figura 8.2: Equipamento Desenvolvido

Para criação deste equipamento, foi usado a carcaça de um switch 3Com SuperStack II Switch 3300. Todos os componentes internos foram removidos e a base metálica foi remodelada e pintada para acomodar a placa eletrônica e o microcontrolador. Na parte interna do aparelho, se encontram o display LCD, um fan usado para o resfriamento dos componentes internos da placa, um relé de estado sólido para o acionamento do ar-condicionado, uma chave seletora da tensão de entrada 110V/220V, a placa de circuito desenvolvida e os conectores para a porta do rack e para a fase do condicionado de ar.

A placa de eletrônica pode ser vista na figura 8.3. Foram impressas as legendas de todos os componentes na parte frontal da placa como pode ser vista na figura 8.4. Também foram necessários usar 6 jumpers na parte traseira da placa, como mostra a figura 8.4.

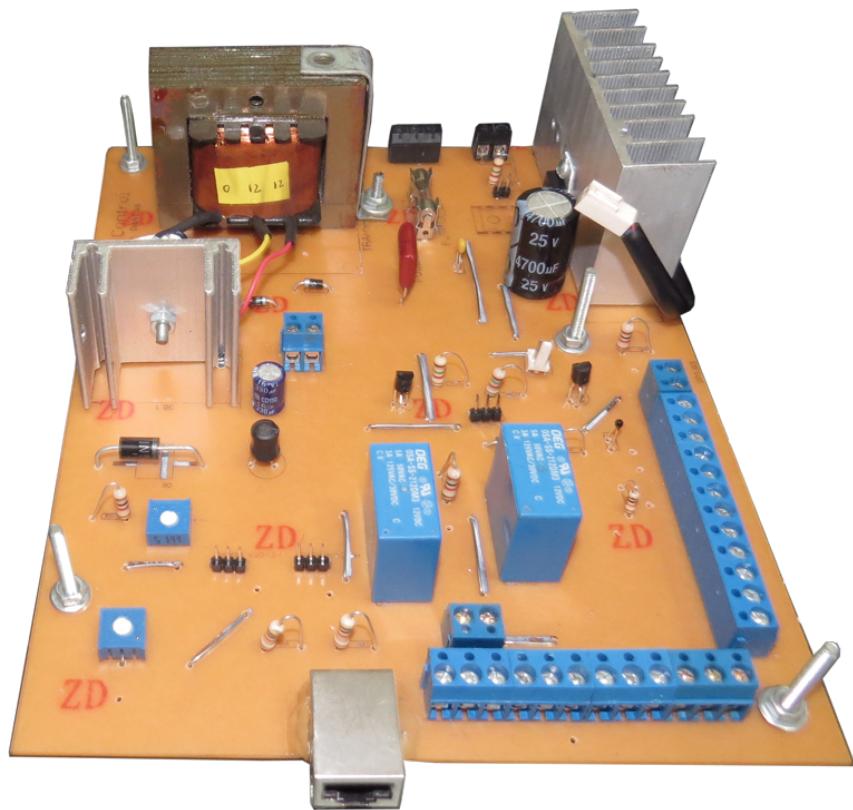


Figura 8.3: Placa Eletrônica Montada

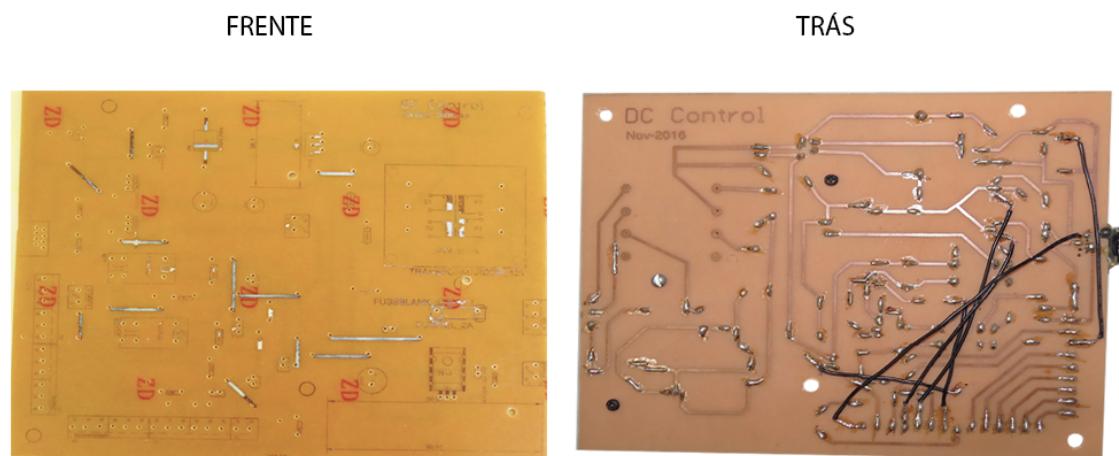


Figura 8.4: Placa de Circuito Impresso Desenvolvida

Antes da confecção da placa de circuito impresso, foi projetado o circuito completo em protoboard como mostrado na figura 8.5, onde todos os equipamentos e dispositivos estão conectados. Uma lâmpada foi usada para simular a fechadura magnética. Esta lâmpada possui o mesmo consumo de corrente da fechadura usada, de 400 mA.

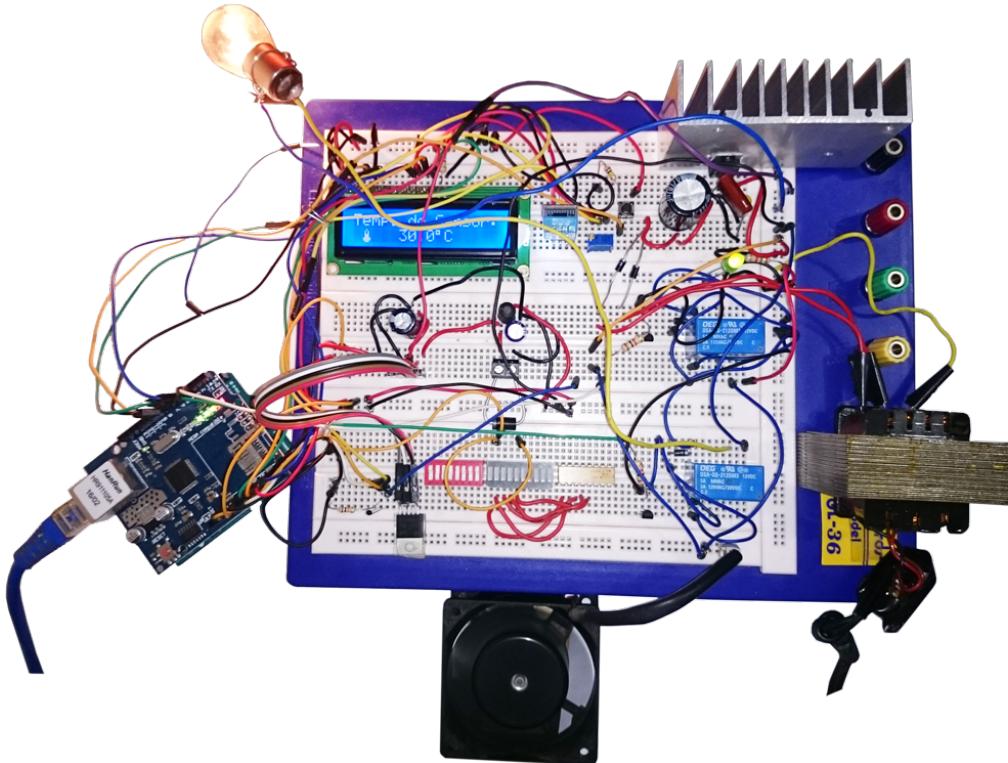


Figura 8.5: Projeto em Protoboard

Um outro equipamento que ainda está em desenvolvimento é mostrado na figura 8.6. Este equipamento esta sendo desenvolvido no PoP-RN e foi usado para simulações e testes. Nesta versão são usados uma fonte de bancada e um switch para conexão do dispositivo na rede. Além disso, o display LCD usado neste modelo é do tipo 20x4, o que torna possível mostrar mais dados.

Além deste equipamento da figura 8.6, outras 10 unidade estão em desenvolvimento pelo PoP-RN. Estes equipamentos serão instalados nos gigaPOPs da empresa e farão o monitoramento do ambiente, controle de acesso a porta do rack e o acionamento do ar-condicionado.

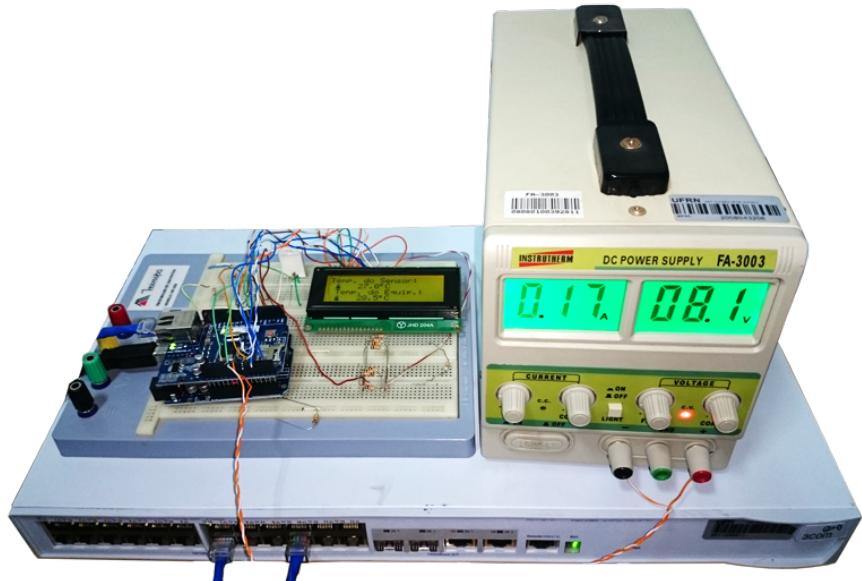


Figura 8.6: Projeto em Desenvolvimento no PoP-RN

O sensor de temperatura e umidade foi projetado com objetivo de facilitar a instalação e manutenção do mesmo. Para isso, o sensor DHT11 foi fixado em uma caixa plástica e um conector RJ45 foi usado para conexão, como mostra a figura 8.7. Além do DHT11, alguns equipamentos usam o sensor DHT22, como é o caso do exemplar em desenvolvimento mostrado na figura 8.6.



Figura 8.7: Sensor de Temperatura Desenvolvido

8.1.1 Análise Elétrica

Foram realizados testes em laboratório do circuito mostrado na figura 8.5 e calculados a potência, consumo e eficiência do equipamento. A fonte linear de 12V apresentou uma variação de apenas 1,6% na variação da tensão de saída quando operando com carga máxima. A curva da tensão de saída é mostrada na figura 8.8. A variação total resultante foi de 192 mV, o que está dentro do esperado pois o circuito foi testado ainda em protoboard.

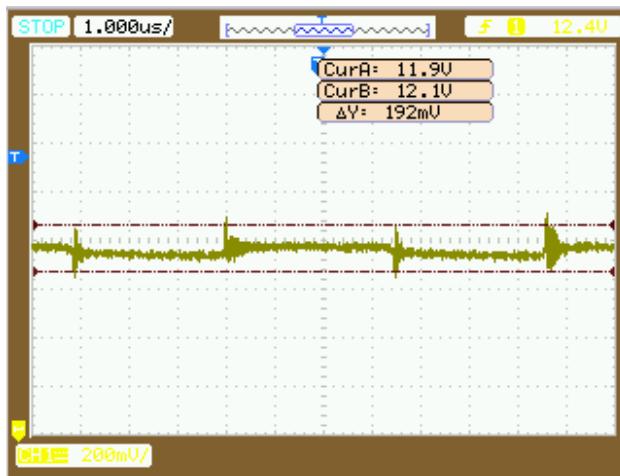


Figura 8.8: Gráfico de Saída da Fonte Linear 12V

A fechadura magnética foi simulada com uma lâmpada de consumo de 400 mA em 12V DC. O Consumo total da equipamento foi de 763 mA em 12V DC. Para o cálculo da potência do equipamento, foi usado a equação 8.1.

$$P_{\text{med}} = \frac{1}{T} \int_0^T I(t) \cdot U(t) \cdot dt \quad (8.1)$$

Onde $I(t)$ é o valor da corrente no instante t e $U(T)$ o valor da tensão no mesmo instante. Portanto, para um instante de tempo T temos que:

$$P_{\text{med}} = I \cdot U \quad (8.2)$$

Logo, obtivemos a partir da equação 8.2 uma potência média de consumo da fonte de 12V de 9,16 W. O fan apresentou um consumo de 123 mA em 12V DC. Os resultados alcançados foram aceitáveis e dentro do esperado.

A fonte buck de 5V apresentou resultados ótimos. A figura 8.9 mostra a saída do conversor buck operando com carga máxima, que nesse caso é representada pelo microcontrolador, Shield Ethernet, display LCD e sensores. A variação encontrada foi de aproximadamente 280mA.

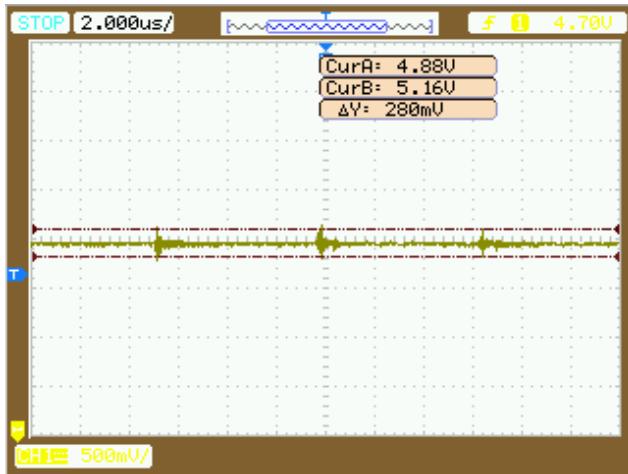


Figura 8.9: Gráfico da Saída da Fonte Buck 5V

A partir da medição do consumo das correntes de cada componente do circuito, foi determinada a eficiência da fonte buck a partir do cálculo da potência elétrica de entrada e saída do circuito. A eficiência é determinada pela equação 8.3.

$$e = \frac{P_{\text{saída}}}{P_{\text{entrada}}} \quad (8.3)$$

A eficiência encontrada foi de aproximadamente 76%. O resultado obtido foi muito promissor pois foi um pouco superior a eficiência estimada pelo fabricante do CI usando para realizar o chaveamento (Texas Instruments Incorporated, 2016).

Umas das razões que ocasionou esta eficiência foi a escolha de um indutor e capacitor com valores um pouco superiores aos sugeridos. Como resultado obtivemos uma fonte sem grandes quedas de tensão quando ocorre um aumento da carga.

8.2 Monitoramento e Controle com o Software DC Control

Foram realizados testes de monitoramento e controle com dois equipamentos mostrados nas figuras 8.2 e 8.6. Uma rede IPv4 foi criada com faixa de IP 192.168.2.0 e máscara de 24 bits. Foi usando um wireless access point de 150 Mbps para testes em uma smartphone executando o sistema operacional Android. Foram usados um servidor dedicado Linux executando a aplicação para coleta de dados Zibbax e um notebook executando o sistema operacional MacOS. As duas estações foram conectadas a um switch 1000BASE da 3Com pois a maioria dos gigaPOPs são equipados com switches de conexão 1000BASE X/T. A figura 8.10 mostra a arquitetura montada.

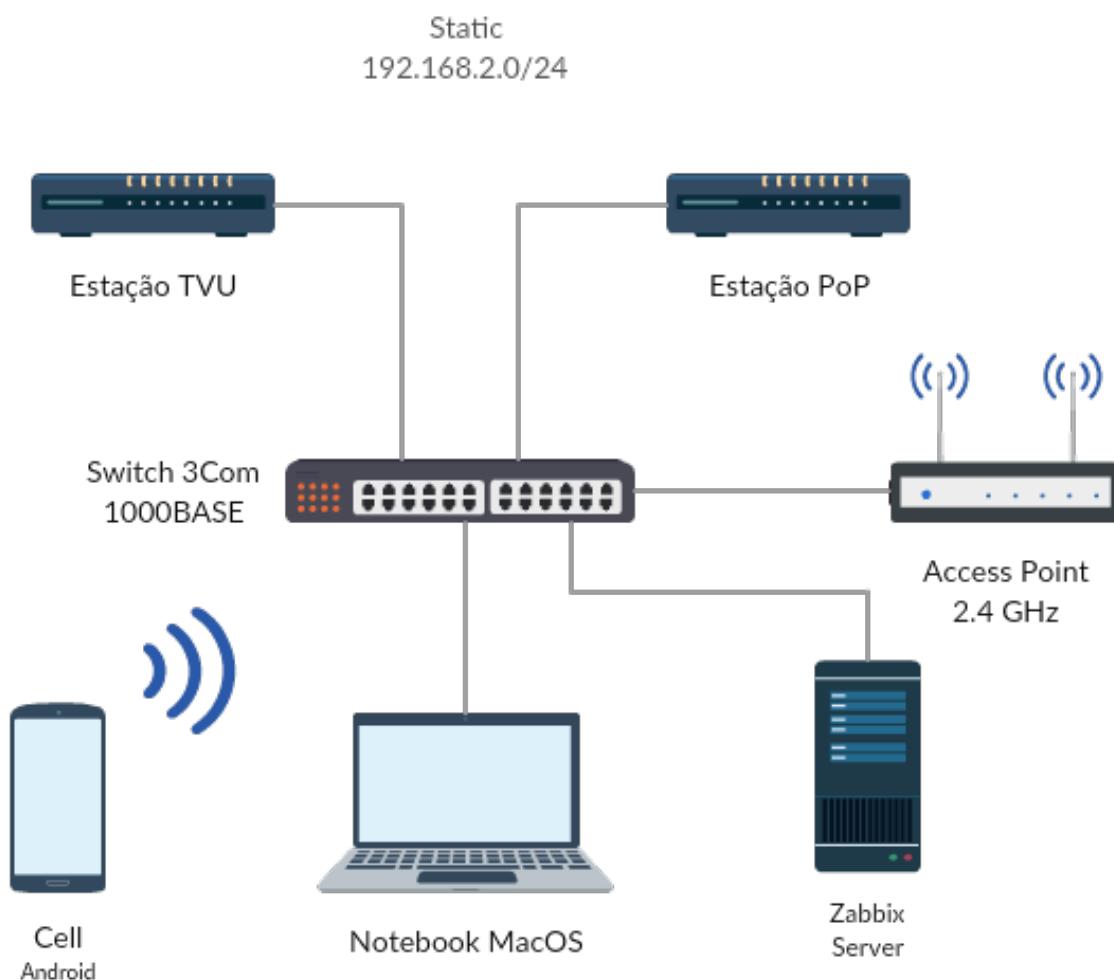


Figura 8.10: Rede Criada para Testes

Foi realizado o monitoramento de um uma sala de computadores por aproximadamente 6 horas. A figura 8.11 mostra o resultado de aproximadamente 15 minutos de um dos testes realizados. Neste teste, os equipamentos foram colocados em zonas diferentes de temperaturas e a temperatura do ambiente foi aumentada gradualmente durante o teste. O gráfico mostra os valores dos sensores do equipamento e sensores externos. Observe que existem algumas perturbações no sensor externo da estação MAIN. Esses erros estão associados a imprecisão do sensor DHT11. Tais erros não acontecem na estação TVU pois ela está equipada com o sensor DHT22.

Um outro teste foi realizado registrando a queda de temperatura do ambiente pela estação TVU. A figura 8.12 mostra o registro da temperatura durante 80 minutos. Podemos perceber que a temperatura teve uma variação de aproximadamente 8° C. Foi usado um termômetro externo para comprovar a variação térmica e o resultado obtido foi satisfatório.

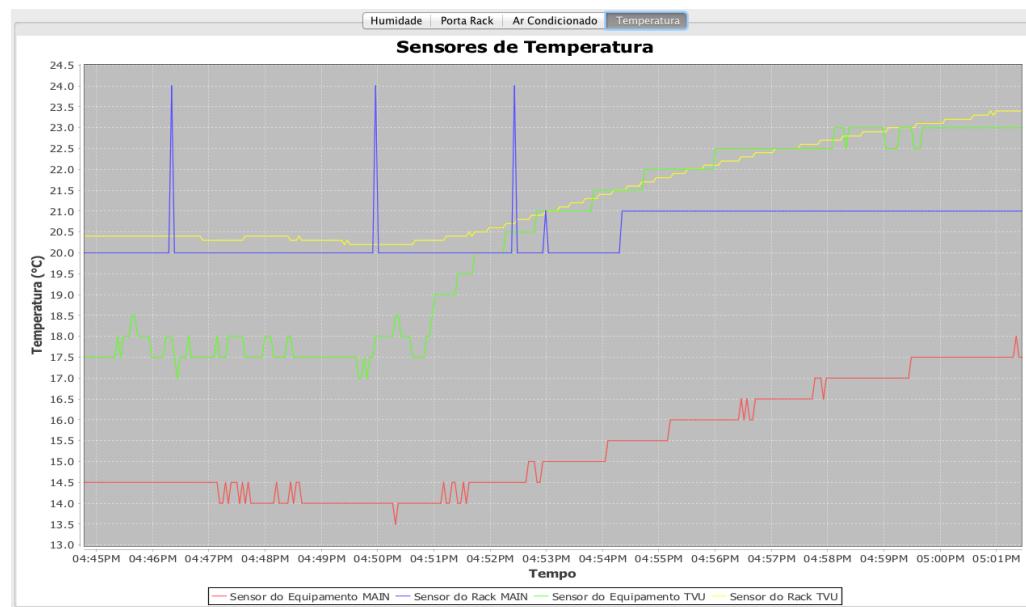


Figura 8.11: Teste de Temperatura das Estações

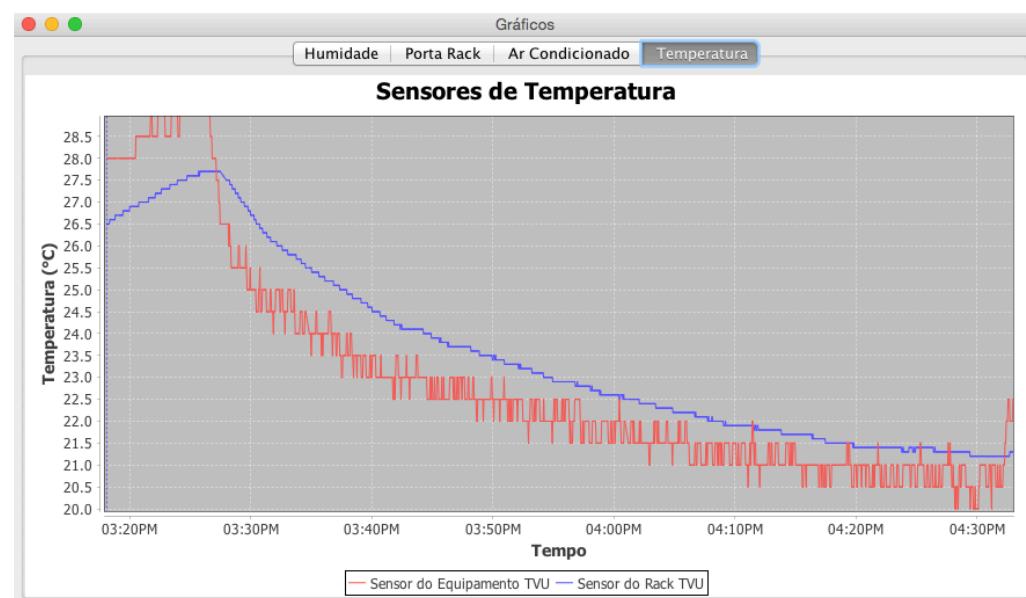


Figura 8.12: Teste de Temperatura Estação TVU

8.3 Monitoramento e Controle por um Smartphone

Existem diversos aplicativos de monitoramento de redes SNMP disponíveis gratuitamente para smartphones Android e iOS. Alguns testes de monitoramento e controle foram realizados usando o aplicativo ¹SNMP MIB Browser para Android. Nesse aplicativo, é possível configurar a OID, porta, comunidade e IP do equipamento que desejamos monitorar. Após configurar o dispositivo, foi possível realizar o monitoramento da temperatura como mostrado na figura 8.13. A temperatura registrada foi de 27°C. Além do monitoramento da temperatura, foi realizado controle da abertura da porta pelo aplicativo com sucesso.

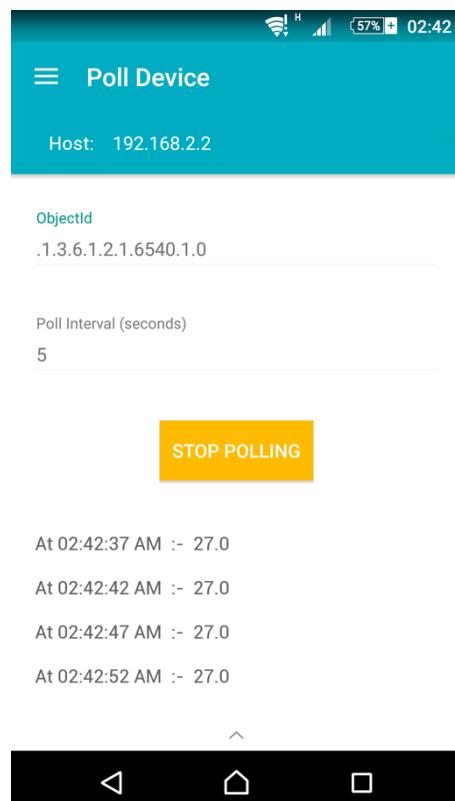


Figura 8.13: Monitoramento da Temperatura pelo Smartphone

¹ ManageEngine SNMP MIB Browser é uma aplicativo proprietário da empresa americana Zoho Corporation que permite o monitoramento e gerenciamento SNMP de dispositivos de redes e possui licença gratuita para algumas de suas funções.

8.4 Monitoramento com o Software Zabbix

Um servidor Unix baseado na distribuição Linux Ubuntu foi usado para a instalação do software de monitoramento Zabbix versão 3.2.3. O servidor foi virtualizado usando o software de virtualização em nuvem VMware vSphere, e possui 2 GB de memória RAM, armazenamento de 20 GB e um único processador. A partir da ferramenta de criação de mapas de redes do Zabbix, foi desenvolvido um mapa que mostra a arquitetura da rede que foi criada para o gerenciamento como mostra a figura 8.14. O mapa mostra o estado da conexão de cada host, assim como também os alertas configurados. Apenas uma estação foi usada para o monitoramento. Além disso, é possível realizar testes de diagnósticos para cada host.

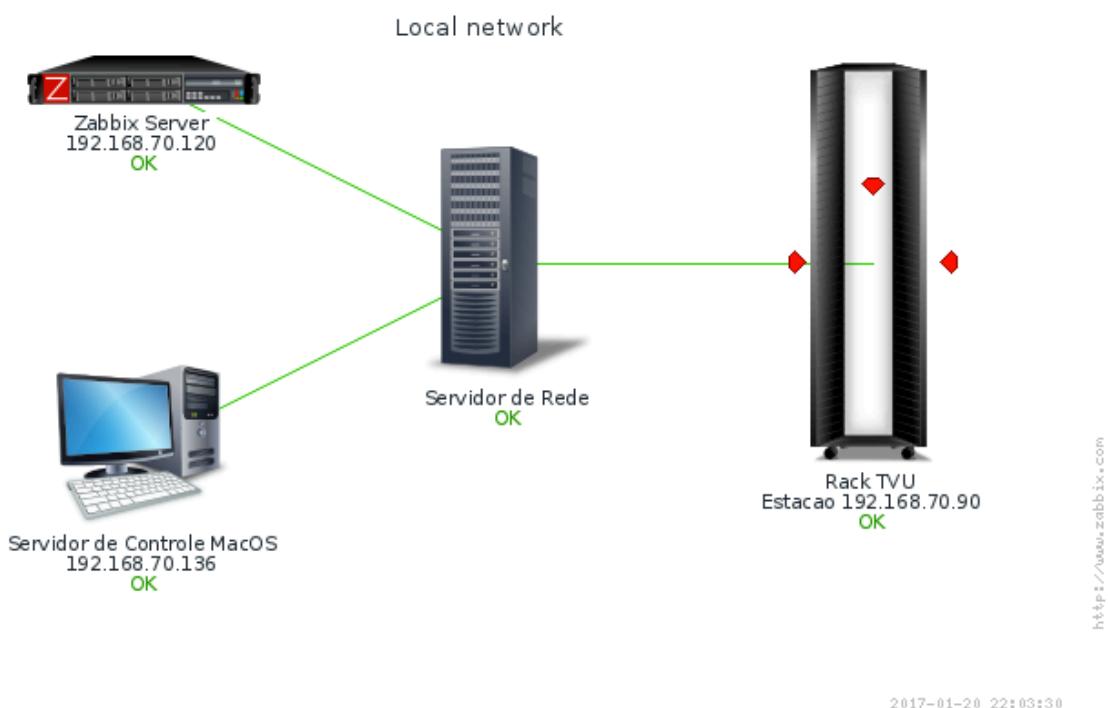


Figura 8.14: Mapa de Rede com o Zabbix

Foi adicionado um host que representa a estação que está sendo gerenciada ao servidor Zabbix com comunicação SNMPv1. Usando a ferramenta de criação de variáveis, foi configurado os parâmetros das 6 variáveis da estação TVU. A figura 8.15 mostra as variáveis que foram criadas.

Name ▲	Inter...	History	Trends	Type	Last check	Last value	Change	Info
- other - (6 items)								
Estado Saidas EstadoSaidas	2	90		SNMPv1 ...	2017-01-17 21:06:59	AR 0 PORTA 1		History
Humidade Humidade	1	90	365	SNMPv1 ...	2017-01-17 21:07:00	43.2 %		Graph
Last Boot Segundos LastBoot	2	90	365	SNMPv1 ...	2017-01-17 21:06:59	742 segundos	+2 segundos	Graph
Nome da Estacao EstacaoName	2	90		SNMPv1 ...	2017-01-17 21:06:59	Estacao_Pop-RN		History
Temperatura do Equipamento TempEquipamento	1	90	365	SNMPv1 ...	2017-01-17 21:07:00	22.1 C	+0.1 C	Graph
Temperatura do Rack TempRack	1	90	365	SNMPv1 ...	2017-01-17 21:07:00	20.9 C		Graph

Figura 8.15: Variáveis Gerenciáveis com o Zabbix

Uma tela contendo todos os gráficos e variáveis de gerenciamento foi montada no Zabbix e é mostrada na figura 8.16. Nesta tela, é possível dar um zoom em uma horário específico no gráfico ou obter o histórico dos dados de último boot do equipamento.



Figura 8.16: Tela de Gerenciamento Zabbix

Para o teste realizado, foi criado um alerta para quando a temperatura do equipamento da estação TVU atingisse a temperatura de 25°C. Quando o *trigger* é acionando, um alarme é criado e um email é enviado alertando o administrador de rede do incidente. Observe no gráfico da figura 8.16 que em 22:00 a temperatura do equipamento atingiu o limite definido e o alarme foi gerado. O email foi enviado para uma conta Gmail configurada no servidor Zabbix. Após a normalização da temperatura do equipamento, um novo email é enviado informando a normalização mostrando o valor atual

da temperatura. O email usa o protocolo Simple Mail Transfer Protocol (SMTP) para enviar a mensagem. A figura 8.17 mostra a mensagem de teste que foi criada e enviada pelo Zabbix.

Time	Action	Type	Recipient	Message	Status	Info
2017-01-19 22:59:52	Report problems to Zabbix administrators	Email Jaime gmail Novo	Admin (Zabbix Administrator) jaimejales@hotmail.com	Subject: ola mundo Message: teste de mensagem de temperatura		Sent

Figura 8.17: Mensagem do Envio de Email do Zabbix

Além da temperatura e humidade de cada gigaPOP, a tela também mostra qual foi o ultimo boot do equipamento e quais são os estados atuais do ar-condicionado e da porta do rack. Também foram configurados alertas para quando o equipamento reiniciar e quando ocorrer perda de conexão. Para esses casos, um email é enviado alertando os técnicos responsáveis. Os resultados obtidos com o Zabbix foram muito satisfatórios e promissores.

Foi comprovado que o equipamento é capaz de manter comunicações simultâneas entre mais de um gerente SNMP, pois o servidor de Controle Java DC Control também ficou ativo durante os testes. Além disso, em nenhum momento o equipamento apresentou travamentos ou falhas na conexão SNMP.

Esta tela apresentada na figura servirá de base para construção de uma tela mostrando todas as temperaturas dos racks das estações que será mostrada em uma TV de 42” para monitoramento visual em tempo real dos gigaPOPs.

Capítulo 9

9 Conclusões

Um sistema embarcado baseado em tecnologias de baixo custo capaz de realizar o acionamento de condicionadores de ar e a abertura de fechaduras magnéticas de portas de rack de comunicação foi proposto neste trabalho. Além disso, o sistema apresentado é capaz de realizar o monitoramento de sensores internos e externos de temperatura e umidade através do protocolo de gerenciamento SNMP.

Em adição a arquitetura de hardware proposta, também foi apresentada um software capaz de monitorar e controlar múltiplas estações de monitoramento. Umas das maiores vantagens do projeto é seu baixo custo de fabricação. Custando aproximadamente R\$ 300,00 o equipamento proposto se enquadra na categoria das soluções mais baratas e eficientes para realizar o monitoramento de data centers e de outros ambientes industriais.

A arquitetura criada é composta por um microcontrolador ATmega328p, um Shield Ethernet W5100, sensores DHT11 e DHT22, e uma placa eletrônica contendo duas fontes de tensão e um circuito de controle. O software de monitoramento DC Control é baseado na linguagem Java e usa o protocolo de gerenciamento SNMP para comunicação com as estações.

9.1 Trabalhos Atuais e Futuros

Atualmente estão sendo desenvolvidos 10 unidades do equipamento proposto neste trabalho. Uma delas, apresentada no capítulo 8, já está em funcionamento em um laboratório de testes localizado no PoP-RN. As unidades restantes estão em fase de aquisição por licitação, o que é um processo demorado que leva alguns meses.

Um servidor executando o software Zabbix já está em funcionamento realizando o monitoramento da estação localizada no PoP-RN. Atualmente, funcionalidades adicionais estão sendo discutidas e moldadas para a implementação nesse sistema. A mais promissora delas que já está em desenvolvimento é a integração do sistema à medição de consumo da rede elétrica e para diagnósticos de falhas elétricas. Sensores estão sendo projetados para a integração ao sistema.

O equipamento proposto nesse trabalho suporta todas as funcionalidades implementadas pelos mais renomeados equipamentos de monitoramento de data centers existentes no mercado. A ideia é que em um futuro próximo seja possível lançá-lo no mercado nacional como uma solução de baixo custo para monitoramento de pequenos e médios data centers.

Referências Bibliográficas

- Aosong Electronics. (2016). *DHT11*. Acesso em 10 de Janeiro de 2017, disponível em Aosong: <https://akizukidensi.com/download/ds/aoSong/DHT11.pdf>
- Aosong Electronics. (2016). *DHT22*. Acesso em 10 de Janeiro de 2017, disponível em Aosong: <http://akizukidensi.com/download/ds/aoSong/AM2302.pdf>
- Araújo, A., & Junior, I. V. (5 de Julho de 2013). Protocolo de Transmissão de Dados SNMP em Redes de Sensoriamento Sobre a Plataforma Arduino. *IX Congresso de Iniciação Científica do IFRN*.
- Arduino. (2016). *Arduino Ethernet Shield*. Acesso em 29 de Dezembro de 2016, disponível em Arduino: <https://www.arduino.cc/en/Main/ArduinoEthernetShield>
- Arduino. (2017). *Arduino Uno*. Acesso em 9 de Janeiro de 2017, disponível em Arduino: <https://www.arduino.cc/en/main/arduinoBoardUno>
- Arduino. (2017). *Millis Function*. Acesso em 5 de Janeiro de 2017, disponível em Arduino: <https://www.arduino.cc/en/reference/millis>
- Atmel Corporation. (2015). *ATmega328P Datasheet*. Acesso em 28 de Setembro de 2016, disponível em Atmel: http://www.atmel.com/images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Complete.pdf
- Beal, V. (2016). *Java*. Acesso em 28 de Dezembro de 2016, disponível em Webopedia: <http://www.webopedia.com/TERM/J/Java.html>
- Born, B. (23 de Janeiro de 2012). *Arduino HTTP Temperature sensor (formerly SNMP)*. Acesso em 5 de Setembro de 2016, disponível em Gridshield: <http://openhardware.gridshield.net/home/arduino-snmp-temperature-sensor>
- Dantas, J., Alsina, P., Júnior, C., Santiago, I., Ramos, J., & Dantas, J. D. (2016). Arquitetura de Hardware Embocado em Carga Recuperável Ejetável para Veículo Aéreo Não Tripulado. *VI Fórum de Pesquisa e Inovação do CLBI* (p. 9). Parnamirim: CLBI.

- FelipeFlop. (2017). *Display LCD 16x2 Backlight Azul*. Acesso em 10 de Janeiro de 2017, disponível em FelipeFlop: <http://www.filipeflop.com/pd-6b7e4-display-lcd-16x2-backlight-azul.html>
- Ferdoush, S., & Li, X. (2014). Wireless Sensor Network System Design using Raspberry Pi and Arduino for Environmental Monitoring Applications. *The 9th International Conference on Future Networks and Communications*, 103-110.
- Fernández, V., Pazos, A., Saborio, J., & Seco, M. (2014). Arduino and Nagios integration for monitoring. *20th International Conference on Computing in High Energy and Nuclear Physics*, 8.
- Fotek. (2016). *DC to AC Solid State Relay*. Acesso em 25 de Janeiro de 2017, disponível em Fotek: <http://www.fotek.com.hk/solid/SSR-1.htm>
- Gionet, E. (2010). *Agentuino Library*. Acesso em 1 de Setembro de 2016, disponível em Google: <https://code.google.com/archive/p/agentuino/wikis/AgentuinoLibrary.wiki>
- Hart, M. (2 de Julho de 2013). *Streaming*. Acesso em 1 de Setembro de 2016, disponível em Arduiniana: <http://arduiniana.org/libraries/streaming/>
- Internet Engineering Task Force. (Maio de 1990). *RFC 1157 - IETF*. Acesso em 1 de Janeiro de 2017, disponível em Internet Engineering Task Force: <https://www.ietf.org/rfc/rfc1157.txt>
- Ipec. (2016). *Fechadura M150*. Acesso em 10 de Janeiro de 2017, disponível em Ipec: http://www.ipec.ind.br/img/produtos/236/M094%20-%20Fechadura%20Magnetika%20-%20M150%20_%20V08.pdf
- ITWatchDogs. (2016). *WATCHDOG 1250 ENVIRONMENT MONITOR*. Acesso em 1 de Janeiro de 2017, disponível em ITWatchDogs: <http://www.itwatchdogs.com/IP-Climate-Monitor-Watchdog-1250-p11.html>
- JFreeChart. (2016). *JFreeChart Home*. Acesso em 1 de Janeiro de 2017, disponível em JFreeChart: <http://www.jfree.org/index.html>
- Machado, L., Mota, A., & Mota, L. (2015). SNMP Management of Urban Areas Remote Monitoring via Open Platform Proxy-IP. *1st International Electronic Conference on Remote Sensing* (p. 6). Basel: MDPI.

- Malesevic, M., & Stupic, Z. (2011). *Thermistor*. Acesso em 20 de Setembro de 2016, disponível em Arduino: <http://playground.arduino.cc/ComponentLib/Thermistor2>
- Mauro, D. R., & Kevin, S. J. (2005). *Essential SNMP* (2. ed. ed.). Sebastopol: O'Reilly Media.
- Microsoft. (28 de Março de 2003). *SNMP Technical Reference* . Acesso em 13 de Janeiro de 2017, disponível em Microsoft: <https://technet.microsoft.com/en-us/library/cc776379%28v=ws.10%29.aspx?f=255&MSPPError=-2147217396>
- Revolution Education. (2016). *Alphanumeric LCD Display (16 X 2)*. Acesso em 10 de Janeiro de 2017, disponível em picaxe: <http://www.picaxe.com/docs/LED008.pdf>
- Romeiro, W., & Costa, F. (2016). Monitoramento Residencial Utilizando o Zabbix e o Padrão IEEE 802.15.4. *32nd HOLOS*. 1, p. 10. Natal: IFRN.
- Santos, J. C. (2009). *Projeto de um sistema microcontrolado utilizando Internet embarcada para monitoramento remoto em tempo real de temperatura e disponibilização dos dados na WEB através de conexão de rede*. Universidade Federal do Rio Grande do Norte, Departamento de Engenharia de Computação e Automação. Natal: UFRN.
- Santos, M., Tarouco, L., Bertholdo, L., Lima, F. M., & Vasconcellos, V. (2015). *Gênica de Redes de Computadores* (2. ed. ed.). Rio de Janeiro: Escola Superior de Redes RNP.
- Schneider Electric. (2016). *APC NetBotz*. Acesso em 2 de Janeiro de 2017, disponível em NetBotz Sensors : http://www.apc.com/shop/us/en/categories/security-and-environmental-monitoring/sensors-cameras-licenses-accessories/netbotz-sensors/_/N-qlim2m
- Sehgal, A., Perelman, V., & Kuryla, S. (24 de Dezembro de 2012). Management of resource constrained devices in the internet of things. *IEEE Communications Magazine*, 50(12).
- Shibu, K. (2009). *Introduction to Embedded Systems* (Vol. 1). New York: Tata McGraw.

- SNMP4J. (2015). *SNMP4J - The Object Oriented SNMP API for Java Managers and Agents*. Acesso em 1 de Janeiro de 2017, disponível em SNMP4J: <http://www.snmp4j.org/index.html>
- Texas Instruments Incorporated. (2016). *LM2596 - Texas Instruments*. Acesso em 21 de Setembro de 2016, disponível em Texas Instruments: www.ti.com/lit/ds/symlink/lm2596.pdf
- Waltronica. (2017). *TERMISTOR*. Acesso em 9 de Janeiro de 2017, disponível em Waltronica: <http://www.waltronica.com.br/subcategoria/1852-termistor>
- Wikipedia. (2 de Novembro de 2016). *NetBeans*. Acesso em 28 de Dezembro de 2016, disponível em Wikipedia: https://en.wikipedia.org/wiki/NetBeans#cite_note-4

Apêndices

APÊNDICE A – Projeto Eletrônico do Equipamento

