

TRABALHO DE GRADUAÇÃO

**AMBIENTES INTELIGENTES PARA RACIONALIZAÇÃO
DE ENERGIA UTILIZANDO LOCALIZAÇÃO RFID**

Josué Fernandes Souza

Ariel Caleb Fernandes Souza

Brasília, março de 2011



**ENGENHARIA
MECATRÔNICA**
UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia

TRABALHO DE GRADUAÇÃO

**AMBIENTES INTELIGENTES PARA RACIONALIZAÇÃO
DE ENERGIA UTILIZANDO LOCALIZAÇÃO RFID**

Josué Fernandes Souza
Ariel Caleb Fernandes Souza

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

Banca Examinadora

Prof. Dr. Adolfo Bauchspiess, ENE/UnB

Orientador

Prof. Dr. Edson Paulo da Silva, ENM/UnB

Examinador externo

Prof. Mestre Jones Yudi Mori Alves da Silva,

ENM/UnB

Examinador externo

Dedicatórias

Dedico esse trabalho a Cristo o autor e consumidor da minha fé, aquele em quem eu confio. Ao meu irmão Josué Fernandes Souza, aos meus pais Paulo Rubem e Neire Maria, aos meus irmãos Jhonatas e Gerson, minha cunhada Natane Souza, aos meus familiares, tios tias e primos. Ao meu amigo Artur Vargas, aos meus colegas de curso, amigos do MCAS e outras denominações, aos meus caríssimos amigos do NVC.

Ariel Caleb Fernandes Souza

Dedico este trabalho ao meu Senhor e Salvador Jesus Cristo. Dedico ao meu querido irmão Ariel Caleb Fernandes Souza, à meus pais Paulo Rubem e Neire Maria, aos meus irmãos Jhonatas, Gerson e Natane e à minha querida e amada namorada Daniele, dedico a toda minha família, dedico também a todos os meus amigos do Núcleo de Vida Cristã.

Josué Fernandes Souza

Agradecimentos

Meu maior agradecimento é a Deus, Ele que é o autor da minha existência, aquele que me sustenta em suas mãos dia após dias, agradeço também aos meus queridos pais Paulo e Neire que sempre me ensinaram a viver uma vida reta e digna, com respeito e honra, amo vocês. Aos meus irmãos Jhonatas e Gerson que sempre me aturaram em meio a esses anos duros de faculdade quero agradecer em especial ao meu irmão, amigo, companheiro, parceiro e ajudante Ariel, sem você nada disso seria possível. Quero agradecer aos meus avós Antônio, Vivi, Rubenil e Alfredo por sempre serem tão carinhosos e atenciosos comigo aos meus tios por serem uma base na qual eu sempre me senti seguro, amo vocês demais, em especial às minhas tias Alessandra, Viviane, Neide e Toninho, amo vocês de uma maneira especial, agradeço também à minha amada namorada Daniele, que sempre foi minha amiga, minha confidente, que mesmo sem saber me ajudou a suportar os dias mais sombrios na Universidade com graça e alegria, te amo além do que podia imaginar. Quero agradecer aos colegas de curso por todos os momentos de ensino e companheirismo, aos meus amigos do Núcleo de Vida Cristã, sem vocês essa jornada seria tenebrosa e solitária, ao lado desses amigos passei os melhores momentos que tive na universidade, aprendi muito com cada um de vocês. Agradeço aos amigos da minha igreja, em especial ao meu amigo Artur.

Quero agradecer em especial ao meu professor e orientador Adolfo Bauchspiess, por todos os conselhos, pelas conversas, pelas horas investidas em mim, sua ajuda e suporte foram de suma importância para minha formação como engenheiro.

Por último quero agradecer a todos os funcionários da SPIN Engenharia de Automação, vocês foram de suma importância na conclusão deste trabalho, não existem maneiras de agradecer o suficiente pelo apoio, pelos conselhos, pela ajuda com cada equipamento e pela confiança depositada em mim e nesse trabalho.

Josué Fernandes Souza

Quero primeiramente agradecer a Deus, por ter me dado forças para eu cursar mecânica, aos meus pais Paulo Rubem e Neire Maria foram esses amigos me dando palavras de ânimo e suporte não somente financeiro, aos meus irmãos Jhonatas e Gerson por sempre me incentivarem e por tantos dias de alegria ao lado deles. A toda minha família, avós, tios, primos etc.

Não Posso me esquecer do NVC, uma família para mim dentro da UnB, lugar onde passei muitas alegrias, momentos inesquecíveis nos picknicks no gramado da reitoria, do ceubinho.

Lembro de um amigo em especial, Artur Vargas, que sempre esteve ao meu lado, valeu mesmo. Agradeço aos meus amigos da igreja ministério Ceifa, local onde congrego e recebo muito.

Ao Professor Adolfo, por seu apoio e por acreditar em meu potencial, mesmo que algumas vezes eu mesmo duvidava, por ceder seu tempo, mesmo de férias para nos auxiliar. Obrigado por tudo.

Por último, quero agradecer aos meus colegas de classe, pelo suporte e apoio, noites em claro, ou até mais tarde no LARA no CA, com trabalhos e provas, creio que não jo teria uma turma melhor para entrar, pessoas unidas e dispostas a ajudar, A galera do LARA, sempre disposta a ajudar e trazer luz nos momentos de escuridão.

Ariel Caleb Fernandes Souza

RESUMO

Há tempos o Laboratório de Automação e Robótica da Universidade de Brasília vem desenvolvendo projetos na área de ambientes inteligentes visando a racionalização de energia e o conforto térmico por meio automação predial. Inserido nesse projeto está o presente trabalho de graduação, seu objetivo é acionar aparelhos de ar condicionado por meio de um sistema supervisorio, decisão essa de ligar ou desligar os aparelhos tomada de acordo com a localização de usuários em um ambiente fechado fazendo-se uso da tecnologia *Radio Frequency Identifier* (RFID) e de redes neurais implementadas e disponibilizadas pelo laboratório. O foco deste trabalho é a comunicação entre o sistema supervisorio e o sistema de localização, a comunicação entre o sistema supervisorio e os aparelhos de ar condicionado e por fim uma tentativa de melhoramento da localização de usuários quando estes estiverem em movimento, dado que o sistema previamente implementado funcionava apenas para os casos em que o usuário estivesse parado. Vários *softwares* e equipamentos foram utilizados, RFID ativos em três ambientes distintos do laboratório, o *software* supervisorio ActionView, o módulo de comunicação ModBus ERS 1050, *scripts* em MatLab para aquisição de dados das leitoras RFID e para treinamento e uso das redes neurais artificiais Perceptron Multicamadas. O Acionamento foi realizado com êxito, quando o supervisorio envia o sinal de liga, ou de desliga, os comandos são executados imediatamente e logo são mostrados na tela do *software*, a comunicação com o sistema de localização está funcionando sem erros através do protocolo *OLE for Process Control (OPC)*, e as redes neurais estão realizando localizações com uma precisão aceitável para ambientes fechados. Contudo faz-se necessário um desenvolvimento melhor na parte de localização para obter-se uma robustez maior para o projeto.

Palavras-chave: supervisorio, *OLE for Process Control (OPC)*, *ModBus*, redes neurais artificiais, filtro de Kalman e ambientes inteligentes.

ABSTRACT

For a long time the Automation and Robotics Laboratory of the University of Brasilia is developing works in Ambient Intelligence, studying ways to save energy while ensuring thermal comfort to the users, through building automation. This graduation work is inserted in this context, its goal is to actuate air conditioning equipments through a supervisory system, this decision of turning on or turning off the equipments is taken according to the localization of the users in the environment, using the Radio Frequency Identifier technology, and neural networks implemented and available at the Laboratory. This work focuses on communication between the supervisory and the localization system, the communication between the supervisory and the air conditioning equipments and a try of improvement of the localization of users when these are moving around, because the localization system that exists works only if the user is static. Many software and equipments were used in this work: active RFID in three rooms of the laboratory, the supervisory software ActionView, the ModBus communication module ERS 1050 [SPIN Engenharia de automação], Matlab scripts for data acquisition of the RFID readers and for the Multilayer Perceptron neural network train and use. The actuation of the air conditioning equipments is considered successfully when the supervisory send the command to turn on and to turn off, the command is executed in real time and immediately after it has been sent, it appears on the software screen. The communication between the localization system and the supervisory is working without errors using the OLE for Process Control Protocol (OPC), and the neural networks are locating the user with a acceptable precision for indoor localization applications, nevertheless it is still necessary to improve the localization to get into a system that is reliable.

Keywords: supervisory system, OLE for Process Control (OPC), ModBus, artificial neural networks and ambient intelligence.

SUMÁRIO

1	INTRODUÇÃO	1
1.1	CONTEXTUALIZAÇÃO	1
1.2	DEFINIÇÃO DO PROBLEMA	3
1.3	OBJETIVOS DO PROJETO	4
1.4	APRESENTAÇÃO DO MANUSCRITO	5
2	CONCEITOS E FERRAMENTAS UTILIZADOS NO PROJETO	6
2.1	SISTEMAS SCADA	6
2.2	ACTIONVIEW	7
2.2.1	GERAÇÃO DA BASE DE DADOS	7
2.2.2	CONFIGURAÇÃO DE TELAS	8
2.3	REDE NEURAL ARTIFICIAL	9
2.3.1	CARACTERÍSTICAS DE UMA REDE NEURAL ARTIFICIAL	9
2.3.2	O <i>Perceptron</i> MULTICAMADAS E O ALGORITMO <i>'backpropagation'</i>	11
3	SUPERVISÓRIO E LOCALIZAÇÃO RFID	12
3.1	INTRODUÇÃO	12
3.2	DESENVOLVIMENTO NO ACTIONVIEW	12
3.2.1	ENVIANDO DADOS VIA OPC	13
3.2.2	CONFIGURAÇÕES NA TELA	14
3.2.3	ESTADO DO AR CONDICIONADO	15
3.2.4	SCRIPT	15
3.3	RFID	17
3.4	LOCALIZAÇÃO	18
3.4.1	RNA-REDE NEURAL ARTIFICIAL PARA <i>tag</i> ESTÁTICA[9]	18
3.4.2	RNA COM AQUISIÇÃO DE DADOS VIA ROBÔ MÓVEL	20
3.5	OPC	21
3.5.1	CONFIGURAÇÃO NO ACTIONVIEW DE PARÂMETROS PARA OPC	21
3.5.2	ENDEREÇAMENTO DOS PONTOS NA TABELA CANAISPEC	23
3.6	AQUISIÇÃO DE DADOS DAS LEITORAS NO MATLAB	23
3.6.1	OPC EM MATLAB	24
3.7	MODBUS	26
3.8	MÓDULOS DE POTÊNCIA	29

3.9	INTEGRAÇÃO FINAL.....	29
4	RESULTADOS DO PROJETO	33
4.1	COLETA DE DADOS DAS LEITORAS.....	33
4.2	REDE NEURAL ARTIFICIAL	33
5	CONCLUSÕES	44
	REFERÊNCIAS BIBLIOGRÁFICAS	46
	ANEXOS.....	48
A	RNA	49
A.1	REDES NEURAIS ARTIFICIAIS.....	49
A.2	ALGORÍTMO <i>Backpropagation</i>	50
A.3	CONSIDERAÇÕES PRÁTICAS SOBRE O ALGORITMO <i>Backpropagation</i>	51
B	COMO CRIAR VARIÁVEIS	52
B.1	SCRIPT ACTIONVIEW	53
C	PUTTY	60
D	CÓDIGO MATLAB	62
D.1	CÓDIGO LEITURA DE DADOS.....	62
E	TABELA DOS PONTOS DE TREINAMENTO DA RNA	66
F	TABELAS RFID	69
G	DESCRIÇÃO DO CONTEÚDO DO CD	72

LISTA DE FIGURAS

1.1	Vista Noturna da Terra a partir de imagens de Satélite, onde os pontos mais iluminados indicam um maior consumo de energia [2].	2
1.2	Imagem de satélite do reservatório da Usina Hidrelétrica de Sobradinho [2].	2
1.3	Gerador de energia Eólica de grande porte, apud [2].	3
1.4	Ilustração de um sistema de geração fotovoltaica de energia elétrica [2].	4
2.1	Ilustração de um Sistema SCADA para controle de nível [3].	7
2.2	Tela principal do <i>Studio Configurator</i> .	8
2.3	O neurônio artificial.	9
2.4	O neurônio artificial com entrada de polarização.	10
2.5	Funções de ativação típicas para uma RNA [4].	10
2.6	Rede Multicamadas [4].	11
2.7	Perceptron Multicamadas (MLP)[4].	11
3.1	Configuração dos parâmetros da variável analógica OPC :POS_X.	15
3.2	Configuração dos parâmetros da variável de estado que aciona o aparelho de ar condicionado.	15
3.3	Foto da leitora <i>Wavetrend</i> , antena e <i>tag</i> ativa [9].	18
3.4	Ilustração de uma rede neural[9].	19
3.5	Localização das Leitoras no Lara.(Pontos vermelhos).	20
3.6	Localização no Lara dos pontos de treinamento da RNA (pontos amarelos [9]).	20
3.7	Configuração dos parâmetros para OPC do ActionView.	22
3.8	Endereçamento dos pontos na tabela CANAISPEC.	23
3.9	Diagrama de blocos para o programa em Matlab de aquisição de dados das Leitoras.	24
3.10	Tela principal do Simulador de Servidor OPC.	25
3.11	Configuração dos parâmetros do TopServer.	26
3.12	Diagrama de blocos do programa em Matlab para comunicação via OPC.	27
3.13	Tipos de ponto no Modbus.	28
3.14	Gravuras do Módulo de comunicação Modbus e suas conexões.	28
3.15	Visualização de todos os módulos do projeto (“painel mímico”).	29
3.16	Ilustração dos ambientes reconhecidos pela rede neural.	30
3.17	Relé T2405Z-M utilizado para acionamento dos aparelhos ar condicionado (Tectronic).	31
3.18	Ar condicionado híbrido utilizado no projeto.	31
3.19	Divisão dos ambientes para acionamento setorizado do ar-condicionado.	32

3.20	ActionView funcionando para o ambiente 1.....	32
4.1	RSSI coletados, para o robô deslocando-se, aquisição de dados a 1,5 segundos.....	34
4.2	Caminho contendo os 91 pontos escolhidos.	35
4.3	Robô móvel aramis.	38
4.4	RSSIs coletados sem presença humana na Leitora 10.	39
4.5	RSSIs coletados com presença humana na Leitora 10.	39
4.6	RSSIs coletados na Leitora 10	40
4.7	RSSIs coletados na Leitora 10.	40
4.8	RSSIs coletados com a <i>tag</i> parada sobre a mesa [9].	41
4.9	Saída X e Y da RNA esperada.	41
4.10	Saída X e Y da RNA treinada com os dados coletados discrepantes entre si.....	42
4.11	Resultado da RNA para um dos dados treinados.....	42
4.12	Trajetória desenhada no chão do LARA.	43
B.1	Configuração da regional no ActionView.....	52
B.2	Configuração dos parâmetros do Sistema.....	52
B.3	Adicionando um grupo ao Sistema	53
B.4	Configuração dos parâmetros do grupo.	53
B.5	Variáveis iniciais do grupo criado.....	54
B.6	Variáveis utilizadas no projeto.	55
B.7	Configuração dos parâmetros da variável digital múltipla.....	56
C.1	Configuração dos parâmetros iniciais do Putty.....	60
C.2	Configuração dos parâmetros intermediários do Putty.	61
C.3	Tela final após conexão estabelecida.	61

LISTA DE TABELAS

1.1	Consumo de Energia por Setor [1].	3
3.1	Variáveis utilizadas no projeto.	13
3.2	Outras variáveis utilizadas no projeto.	14
4.1	Dados coletados com o robô ARAMIS em movimento.	36
4.2	Dados coletados para um ponto em X e Y	37
E.1	Tabela dos pontos X,Y de treinamento da RNA	66
E.2	Tabela dos pontos X,Y de treinamento da RNA, continuação	67
E.3	Tabela dos pontos X,Y de treinamento da RNA, continuação 2	68
F.1	Lista de comandos fornecido pela Wavetrend	69
F.2	Representação dos dados respondidos	70
F.3	Representação dos dados respondidos-continuação	71

LISTA DE SÍMBOLOS

Símbolos Latinos

A	Área	[m ²]
C_p	Calor específico a pressão constante	[kJ/kg.K]
h	Entalpia específica	[kJ/kg]
\dot{m}	Vazão mássica	[kg/s]
T	Temperatura	[°C]
U	Coefficiente global de transferência de calor	[W/m ² .K]
C	Calor perdido por convecção	[W/m ²]
E_{dif}	Calor perdido por difusão de vapor de água através da pele	[W/m ²]
E_{rl}	Calor perdido por respiração latente	[W/m ²]
E_{rs}	Calor perdido por respiração sensível	[W/m ²]
E_{trans}	Calor perdido por transpiração	[W/m ²]

Símbolos Gregos

α	Difusividade térmica	[m ² /s]
Δ	Varição entre duas grandezas similares	
ρ	Densidade	[m ³ /kg]

Subscritos

amb	ambiente
ext	externo
in	entrada
ex	saída

Sobrescritos

\cdot	Varição temporal
$-$	Valor médio

Siglas

LARA	Laboratório de Autmoação de Robótica
RFID	Radio Frequncy IDentification
OPC	OLE for Process Controle
DLL	Dynamic-Link Library
PROCEL	Programa Nacional de Conservação de Energia Elétrica
IHM	Interface Homem-Máquina
IEEE	Institute of Electical and Eletronics Engineer
ISO	International Organization for Standardization
USB	Universal Serial Bus
UnB	Universidade de Brasília
ANEEL	Agência Nacional de Energia Elétrica
PMV	Predicted Mean Vote
UR	Umidade relativa do ar
OLE	Object Linking and Embedding

Capítulo 1

Introdução

Nos dias atuais, a preservação do meio ambiente encontra-se em voga não somente por estar na mídia, mas devido às mudanças climáticas perceptíveis em todo o mundo como o aquecimento global, o degelo das calotas polares, a redução ou o aumento de chuvas em determinadas localidades.

1.1 Contextualização

A energia, nas suas mais diversas formas, é indispensável à sobrevivência da espécie humana. E mais do que sobreviver, o homem procurou sempre evoluir, descobrindo fontes e formas alternativas de adaptação ao ambiente em que vive e de atendimento às suas necessidades. Dessa forma, a exaustão, escassez ou inconveniência de um dado recurso tendem a ser compensadas pelo surgimento de outro(s). Em termos de suprimento energético, a eletricidade se tornou uma das formas mais versáteis e convenientes de energia, passando a ser recurso indispensável e estratégico para o desenvolvimento socioeconômico de muitos países e regiões [2], como ilustrado pela figura 1.1.

O que antes era inviável para algumas pessoas, como poder trabalhar em uma certa localidade devido às condições climáticas e outros fatores relevantes, hoje em dia tem se tornado algo corriqueiro, graças às inovações tecnológicas que procuram a cada dia proporcionar maior conforto, segurança, versatibilidade expandindo os horários possíveis para se trabalhar, fazer atividades de lazer, culturais e sociais. A energia elétrica tem uma grande importância nesse cenário novo do "mundo moderno", proporcionando melhorias no ambiente onde a maioria dos usuários dessas tecnologias se encontram. Em contrapartida tem-se o fator meio ambiente, que está sendo cada vez mais desgastado com alguns tipos de geração de energia comumente utilizados como hidroelétrica, queima de combustíveis fósseis e nuclear dentre outras.

A Energia Hidroelétrica, historicamente, exigiu a formação de grandes reservatórios e, conseqüentemente, a inundação de grandes áreas. Alguns dos lugares inundados eram áreas produtivas e (ou) de grande diversidade biológica, o que fez com que muitas pessoas e animais fossem realocados. Esses reservatórios criados pelas hidroelétricas, como por exemplo o de Sobradinho no Rio

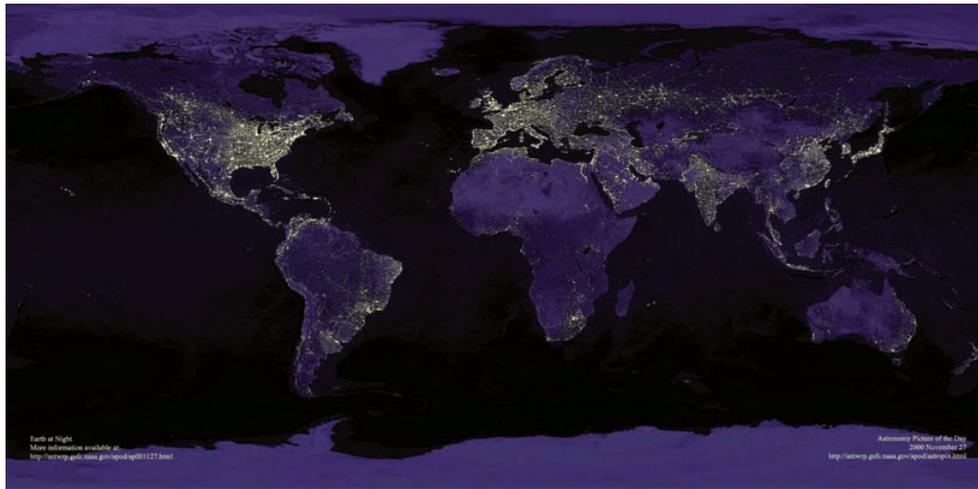


Figura 1.1: Vista Noturna da Terra a partir de imagens de Satélite, onde os pontos mais iluminados indicam um maior consumo de energia [2].

São Francisco (Figura 1.2) provocam alterações no regime das águas e a formação de microclimas, podendo inclusive extinguir algumas espécies nativas. Estima-se que a área inundada por aproveitamentos hidrelétricos no Brasil seja da ordem de 36.000 km^2 [2], o equivalente a 82% da extensão territorial do Estado do Rio de Janeiro e 0,4% de todo o território brasileiro.

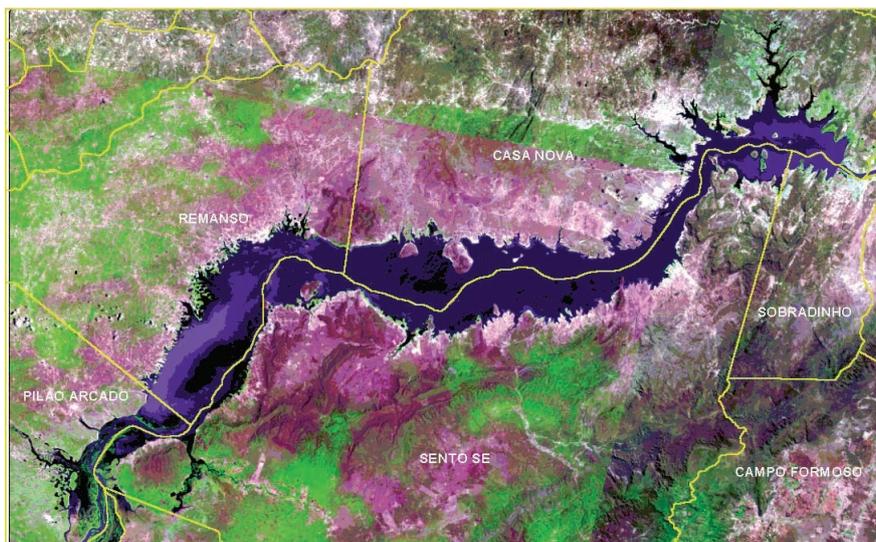


Figura 1.2: Imagem de satélite do reservatório da Usina Hidrelétrica de Sobradinho [2].

Nos últimos anos, o quadro sócio-econômico brasileiro tem apresentado um grande crescimento da demanda de energia elétrica, exigindo uma expansão da potência elétrica instalada a curto e médio prazo, além da adoção de medidas que permitam a conservação de energia. A Tabela (1.1) detalha o perfil do consumo de energia elétrica geral no Brasil. Percebe-se que o setor industrial é o maior consumidor de energia, respondendo por 42% do consumo total. O setor de análise do presente trabalho, que é o comercial e residencial, responde por 43% do consumo de energia elétrica do país [1], ver tabela 1.1- Consumo de energia elétrica por setor. Sabe-se que 48% da

energia gasta no setor comercial é devido aos equipamentos condicionadores de ar, mostrando que a fatia mais relevante do consumo de energia são os sistemas de condicionamento de ar. Dessa parcela, o motor responsável pelo acionamento do compressor consome cerca de 90% da energia do equipamento e o restante é utilizado pelo ventilador. Em sua maioria, os fabricantes de equipamentos condicionadores de ar divulgam valores aproximados de consumo. Para uma mesma carga térmica, o consumo varia de acordo com o tipo de compressor e refrigerante usados para a mesma capacidade de refrigeração. Devido ao crescente problema energético, o PROCEL recomenda uma análise do desempenho térmico das edificações, com objetivo de reduzir sua carga térmica a um valor mínimo[1].

Tabela 1.1: Consumo de Energia por Setor [1].

Setor	Consumo (%)
Rural	15
Comercial e Residencial	43
Industrial	42

Com o alto consumo de energia elétrica e seus imensos benefícios à sociedade, é quase que impossível que se viva sem energia elétrica, porém pode-se pensar em algumas alternativas inteligentes para reduzir o consumo elétrico e diminuir o desperdício. Alguns exemplos já foram citados acima, mas existem muitas outras formas de se economizar energia e evitar desperdícios. Algo que tem se pensado muito é em formas alternativas de geração de energia que tenham menor impacto ambiental como a energia eólica (Figura 1.3), solar (Figura 1.4) e pequenas hidroelétricas que não necessitem de grandes inundações para suprir povoados isolados e algumas fazendas [2].



Figura 1.3: Gerador de energia Eólica de grande porte, apud [2].

1.2 Definição do problema

Ao entrar em um ambiente de trabalho ou domiciliar, as pessoas buscam um certo conforto para realizarem suas atividades corriqueiras, sejam elas relevantes ou não. Em termos de conforto

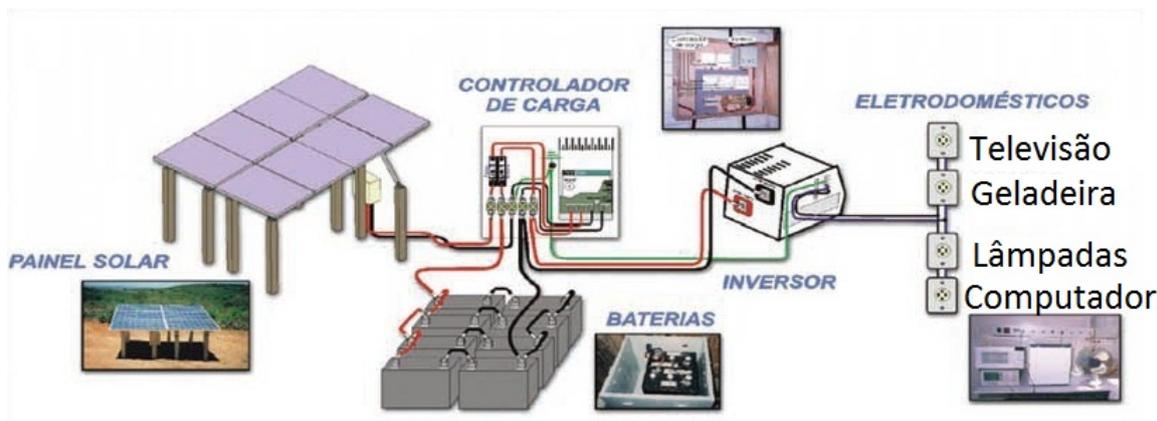


Figura 1.4: Ilustração de um sistema de geração fotovoltaica de energia elétrica [2].

térmico, o aparelho de ar-condicionado tem-se tornado uma solução corrente. O que em sua utilização é o consumo elevado de energia elétrica, e às vezes o uso desnecessário do mesmo, quando pessoas não se encontram no ambiente.

No ambiente onde o projeto está sendo implementado (LARA), foi proposto que o processo de acionamento e desacionamento dos sistemas de ar condicionado fosse feito de acordo com a presença ou não de pessoas no ambiente, o que poderia ser feito com sensores infra-vermelho, porém, as possibilidades de implementação ficariam muito limitadas, pois quando as pessoas estão paradas o sensor infra-vermelho não os detecta dentre outras limitações. Pensando nisso a proposta foi de se realizar uma localização *indoor* de pessoas, através de TAGS ativas (Capítulo 3.3), e então utilizando um supervisor ActionView (Capítulo 2.2) a automação pudesse ser feita, não somente de uma parte do laboratório, mas podendo ser aplicada a outros ambientes se assim for possível, gerando um banco de dados compartilhado com outros aplicativos, sem prejuízo aos pacotes enviados e com o menor intervalo de tempo para processamento e envio dos mesmos.

1.3 Objetivos do projeto

O projeto tem por objetivo acoplar em um *software* SCADA ActionView um sistema de localização de pessoas através de sistemas RFID (Radio-Frequency IDentification) por redes neurais desenvolvido por um projeto anterior [9], esta localização facilitará o controle de temperatura e umidade do ar.

- Comunicação entre o supervisor e o módulo de leitura de dados de *tags* RFID e entre o supervisor e o módulo atuador para o acionamento de aparelhos de ar condicionado;
- Implementação no supervisor da visualização da movimentação de pessoas no ambiente de testes;
- Disponibilização dos dados das TAGS via OPC para diferentes projetos e aplicações;

- Acionamento do aparelho de ar condicionado quando localizada a presença de alguém no ambiente;
- Localização de usuários no ambiente de testes.

1.4 Apresentação do manuscrito

No capítulo 2 é feita uma revisão bibliográfica sobre algumas áreas relacionadas ao assunto abordado no trabalho. Primeiramente apresenta-se o índice PMV de conforto térmico, que é tomado como base para o projeto, pois toda a lógica de ambientes inteligentes utilizada nesse e nos projetos antecessores foram baseados nesse índice. Em seguida é esplanado sobre sistemas SCADA, que em grande parte do projeto será utilizado como supervisor do sistema implementado, também apresenta-se nesse capítulo sobre alguns assuntos estudados pelo grupo para tentativa de implementação do sistema. Em seguida, o capítulo 3 descreve a metodologia empregada no desenvolvimento do projeto, mostrando basicamente cada passo necessário para que a implementação do mesmo fosse feita. Fala-se sobre como configurar o ActionView para o devido projeto. É descrito o funcionamento do RFID e como foi feita a localização de pessoas *indoor* através de Rede Neural Artificial, explicitando qual tipo de RNA, como foi utilizada e treinada e outras informações importantes. No projeto vários programas foram utilizados para auxílio, implementação e estruturação do mesmo, onde cada um deles é especificado para que o leitor possa se situar de forma clara sobre todos os âmbitos e especificações do mesmo. Especificações sobre o uso do Matlab para comunicação OPC e coleta dos dados das leitoras também são descritos no presente capítulo. Resultados experimentais são discutidos no capítulo 4, sobre os dados obtidos das leitoras, a rede neural implementada e seu funcionamento, o acionamento do ar condicionado. No capítulo 5 fala-se sobre os resultados obtidos analisando os objetivos propostos e se foram ou não implementados. Os anexos contêm material complementar ao projeto, alguns algoritmos que podem ser úteis para quem quiser dar continuidade no projeto, como foram feitas as configurações do ActionView, o acesso remoto aos dados e algumas tabelas utilizadas no projeto.

Capítulo 2

Conceitos e ferramentas utilizados no projeto

Serão apresentadas nesse capítulo as principais ferramentas utilizadas para um bom desempenho do trabalho.

2.1 Sistemas SCADA

Originalmente desenvolvido para o monitoramento e controle de chão de fábrica, SCADA é a sigla em inglês para Supervisory Control And Data Acquisition ou controle supervísório e aquisição de dados. Em suas aplicações iniciais, o sistema não era utilizado para processos em tempo real, mas isso vem sendo mudado e já é bastante utilizado para aplicações em tempo real [7]. É utilizado nas áreas de processos industriais, de infra-estrutura e ambientes prediais. O ultimo caso se aplica para o presente projeto.

Um sistema SCADA é geralmente composto pelos seguintes subsistemas:

- IHM (interface homem máquina), tela de apresentação onde uma pessoa é capaz de monitorar e acompanhar o processo e também alterar as referências;
- Um computador supervísório, tem a função de adquirir dados, processar e enviar os comandos necessários para o controle do processo;
- (RTU) Remote Terminal Units ou unidades de terminais remotos, são responsáveis pela conexão com os sensores e pela conversão de sinais analógicos para digitais e pelo envio ao sistema supervísório;
- Infra-estrutura de comunicação conectando todos os pontos de unidade remota com o supervísório.

A figura 2.1 ilustra o uso de um sistema SCADA em um ambiente fabril de controle de nível, utilizando CLPs como RTU.

No projeto de ambientes inteligentes do LARA já é utilizado o software SCADA ActionView. Este foi desenvolvido pela Spin Engenharia e já foi utilizado para os primeiros modelos deste projeto

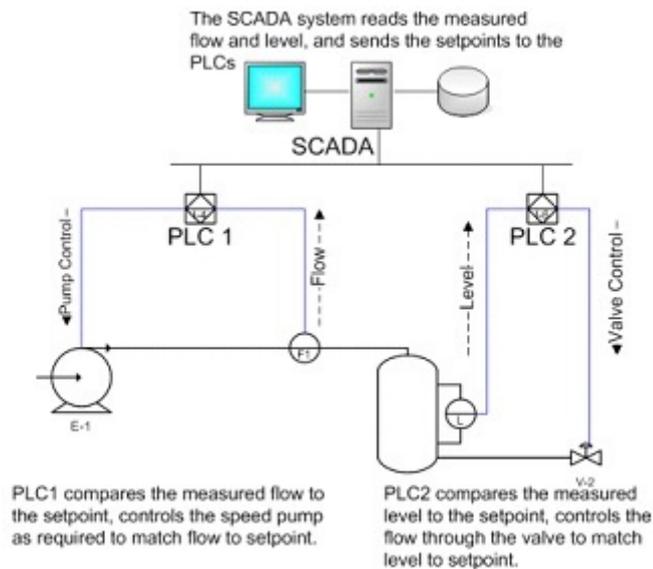


Figura 2.1: Ilustração de um Sistema SCADA para controle de nível [3].

com implementação do sistema na própria Spin engenharia [10]. Na seção 2.2 é apresentado o sistema ActionView e algumas das características deste *software* úteis no projeto.

2.2 ActionView

O Modulo Configurador ActionStudio é um dos aplicativos do sistema que pode ser utilizado *off-line*, durante a fase de projeto, ou *on-line*, durante a execução da supervisão e controle em tempo real. Ele permite que o usuário faça a geração e manutenção do banco de dados de parâmetros do software SCADA, bem como a configuração das telas de processos.

2.2.1 Geração da Base de Dados

Através do modulo *DataBase* são possíveis a inclusão, exclusão e alteração de registros nas tabelas contidas na base de dados paramétrica do sistema. Algumas destas tabelas são: unidades de engenharia, equipamentos, locais, grupos, variáveis calculadas, pontos monitorados, condições para alarmes, cadastro de operadores, senhas, perfis de níveis de acesso, sistemas, regionais (agrupamento de sistemas) etc.

A base de dados organiza os pontos a serem supervisionados e/ou controlados, segundo uma hierarquia de quatro níveis cujo conteúdo é definido inicialmente como: Regional / Sistema / Grupo / Variável.

As variáveis que representam os pontos monitorados e/ou controlados podem ser de entrada, saída ou calculadas analogicamente (medições) e digitalmente. É possível, ainda, criar condições lógicas para alarmes ou para execução de comandos, e também definir funções para cálculos de

pontos não lidos do campo.

O aplicativo dispõe de facilidade para duplicação automática de pontos monitorados que se repetem ao longo do processo supervisionado a partir da definição de um grupo destes pontos. Tal facilidade, denominada 'Grupos Derivados', permite a duplicação de variáveis, funções de cálculo, condições lógicas etc.

2.2.2 Configuração de telas

O aplicativo *Studio Configurator* (parte do software *Actionview*) permite a configuração das telas que serão visualizadas em tempo real (figura 2.2). Para isto, o ambiente integrado e totalmente visual, orientado a objetos, possui uma barra de ferramentas com os tipos de objetos de visualização que podem ser implementados para animação das telas. O usuário pode criar tantas telas de processo ou de tabelas, quantas desejar, e definir regras de navegação entre elas usando botões.

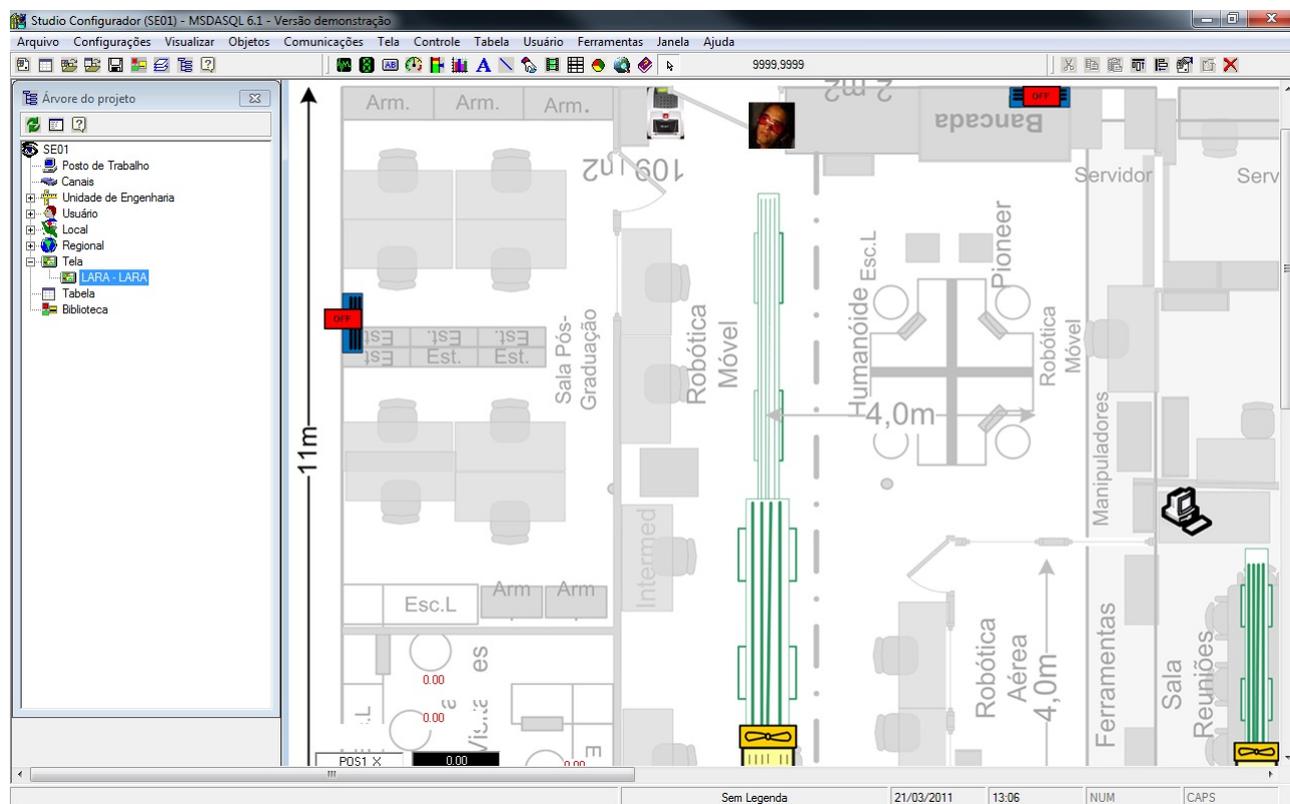


Figura 2.2: Tela principal do *Studio Configurator*.

As telas de processo terão, basicamente:

- Um fundo estático que pode ser um arquivo gráfico padrão (*.BMP, *.DIB, *.EMF, *.JPG, *.PCX, *.WMF e *.ICO), gerado por qualquer editor gráfico;
- Uma parte animada composta por objetos que permitem visualizar o estado / valor de variáveis do processo controlado;

- Uma parte de janelas de comandos ativadas através dos métodos associados aos objetos que permitem a navegação entre telas e a ativação de janelas de ajuda.

Uma tela de processo pode ter muitos objetos de visualização. Para a criação destes objetos e sua movimentação até o local desejado na tela, os objetos são posicionados através de operação de arraste executada com o mouse (*drag and drop*).

As telas de medidas são constituídas por tabelas cujas células podem ser textos constantes, cabeçalhos, valores atuais de variáveis analógicas, valores máximos ou mínimos diários, atualizados em tempo real. Mais especificações de montagens de telas e suas devidas configurações serão apresentadas no Capítulo 3.2 .

2.3 Rede Neural Artificial

Para compreender melhor as redes neurais artificiais é necessário descrever alguns conceitos.

2.3.1 Características de uma rede neural artificial

As redes neurais artificiais podem ser classificadas quanto a:

- Micro-Estrutura - características de cada neurônio na rede;
- Meso-Estrutura - organização da rede;
- Macro-Estrutura - associação de redes eventualmente com processamento analítico para abordar problemas complexos.

2.3.1.1 Micro-estrutura

Cada neurônio artificial possui um número n de entradas cuja soma ponderada pelos pesos w_i passa pela função de ativação para gerar a saída do neurônio. Os pesos w_1, w_2 até w_n (figura 2.3) são variáveis apenas na fase de treinamento. Depois desta fase o neurônio passa a ser uma função não-linear de $\mathcal{R} \rightarrow \mathcal{R}^n$

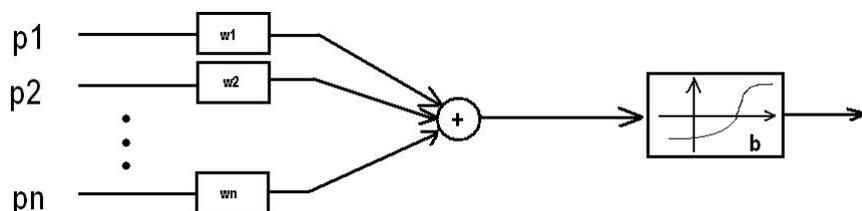


Figura 2.3: O neurônio artificial.

Uma entrada de polarização é freqüentemente utilizada, permitindo uma soma ponderada das entradas não-nula quando a soma das entradas é zero. O parâmetro b (figura 2.3), de deslocamento da função de ativação, torna-se assim desnecessário (figura 2.4).

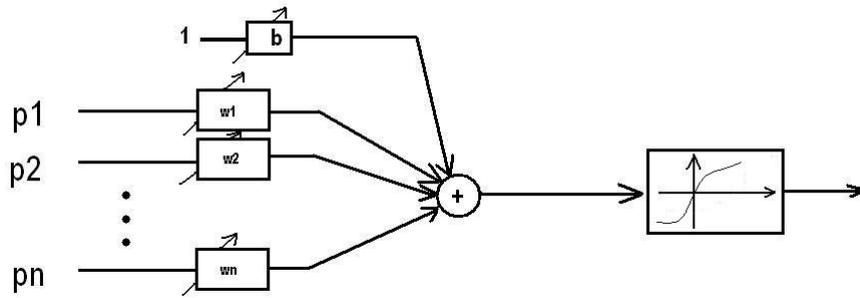


Figura 2.4: O neurônio artificial com entrada de polarização.

Algumas funções de ativação típicas podem ser observadas na figura 2.6.

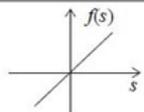
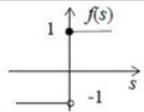
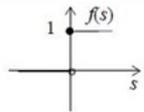
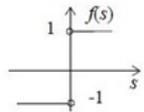
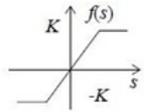
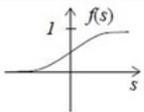
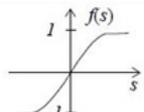
Linear	$f(s) = s$	Hopfield BSB	purelin	
Sinal	$f(s) = \begin{cases} +1 & \text{se } s \geq 0 \\ -1 & \text{se } s < 0 \end{cases}$	Perceptron	hardlims	
Degrau	$f(s) = \begin{cases} +1 & \text{se } s \geq 0 \\ 0 & \text{se } s < 0 \end{cases}$	Perceptron BAM	hardlim	
Hopfield/ BAM	$f(s) = \begin{cases} +1 & \text{se } s > 0 \\ -1 & \text{se } s < 0 \\ \text{inalterado} & \text{se } s = 0 \end{cases}$	Hopfield BAM		
BSB ou Limiar Lógico	$f(s) = \begin{cases} -K & \text{se } s \leq -K \\ s & \text{se } -K < s < +K \\ +K & \text{se } s \geq +K \end{cases}$	BSB	satlin satlins	
Logística	$f(s) = \frac{1}{1 + e^{-s}}$	Perceptron Hopfield BAM, BSB	logsig	
Tangente Hiperbólica	$f(s) = \tanh(s) = \frac{1 - e^{-2s}}{1 + e^{-2s}}$	Perceptron Hopfield BAM, BSB	tansig	

Figura 2.5: Funções de ativação típicas para uma RNA [4].

Meso estrutura Meso-Estrutura - organização da rede observado na figura 2.6.

- Neurônios por camada;
- Camadas da rede;
- Tipo de conexão (*forward*, *backward*, *lateral*).

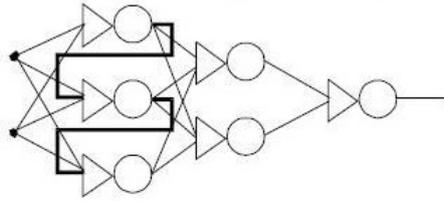


Figura 2.6: Rede Multicamadas [4].

2.3.2 O Perceptron Multicamadas e o algoritmo 'backpropagation'

A figura 2.7 representa uma rede neural perceptron multicamadas com 3 neurônios na camada intermediária e 2 neurônios de saída, essa rede neural pode ser treinada com o algoritmo *backpropagation* utilizando-se o *Pattern P* e o *Target T* para aproximar a saída de funções não lineares.

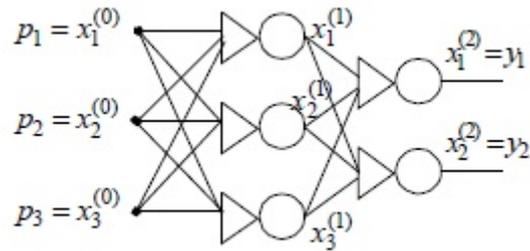


Figura 2.7: Perceptron Multicamadas (MLP)[4].

Capítulo 3

Supervisório e localização RFID

Neste Capítulo estão descritos os vários programas e rotinas necessárias para se realizar a automação pretendida, explicitando quais configurações são necessárias e o que foi feito para alcançar esses objetivos.

3.1 Introdução

Para implementar localização via RFID no ActionView, devem ser definidos alguns parâmetros. É necessário enviar e receber dados das leitoras pela rede, como especificado na seção 3.3.

Para realizar aquisições das leitoras RFID pode ser utilizado um programa no Matlab, que irá receber esses dados e tratá-los de acordo com as especificações da Tabela F, separando-os nos valores de RSSI, intervalo de envio de dados, horário do envio, para verificar se todos os dados chegaram sem serem corrompidos ou extraviados. O ActionView, através de um *SERVER OPC* e posteriormente o Matlab implementa a rede neural, realizando assim a localização da TAG. O MatLab após processar essas informações envia para o ActionView os valores de X, Y e AMBIENTE da *tag* para implementação das rotinas de automação pretendidas para o ambiente.

Os sinais de atuação são enviadas pelo ActionView, através de vários protocolos e saídas diferentes, que permitem uma versatilidade grande quanto à automação.

3.2 Desenvolvimento no Actionview

Para criar as variáveis no ActionView, primeiro é preciso saber ao certo quais são os pontos de atuação, o que será variável interna, quais serão as externas, como pretende-se usar essas variáveis, se serão amostradas na tela do supervisório ou apenas para uso interno.

As variáveis utilizadas podem ser visualizadas na Tabela 3.1, onde as variáveis externas são utilizadas na comunicação com o relé ESR-1050 (figura 3.14) via ModBus, para atualização do estado em que se encontra o ar condicionado, se ligado ou desligado. As variáveis internas são utilizadas para comandos de acionamento e para atualização da posição da *tag* no ambiente. Foram

criadas variáveis de atuação para todos os ambientes, mesmo o trabalho efetivo sendo realizado nos ambientes 1,2 e 3 apenas (figura 3.19).

Tabela 3.1: Variáveis utilizadas no projeto.

Nome	Classificação
ED1-Entrada Digital 1	Externa
ED2-Entrada Digital 2	Externa
ED3-Entrada Digital 3	Externa
ED4-Entrada Digital 4	Externa
ED5-Entrada Digital 5	Externa
ED6-Entrada Digital 6	Externa
ED7-Entrada Digital 7	Externa
ED8-Entrada Digital 8	Externa
SD1-Saída Digital 1	Externa
SD2-Saída Digital 2	Externa
SD3-Saída Digital 3	Externa
SD4-Saída Digital 4	Externa
SD5-Saída Digital 5	Externa
SD6-Saída Digital 6	Externa
SD7-Saída Digital 7	Externa
SD8-Saída Digital 8	Externa

As variáveis EDi e SDi são de retorno para verificação do estado do aparelho de ar condicionado. As variáveis SDi_D e SDi_L são de envio de comando de atuação para o aparelho de ar condicionado. POS_X e POS_Y são as variáveis que armazenam o valor de X e Y enviados ao Actioniew e as variáveis AMBi são para referenciar no script cada ambiente no local de teste do projeto(LARA)

3.2.1 Enviando dados via OPC

Para que possamos ter acesso aos dados do ActionView via OPC, é necessário configurar o arquivo .INI para permitir que o supervisorie envie as variáveis via OPC. Depois de modificado o arquivo, basta fazer uma leitura dos dados OPC do ActionView que ele já envia todas as variáveis criadas no projeto, porém não permite que se escreva nessas variáveis, para tal é necessário que configuremos algumas variáveis que permitam tal operação, conforme a figura 3.1 com o Módulo de comunicação OPC a entrada do tipo analógica (EA) e Endereço1 igual a 1 e Endereço2.

Através do programa TOPSERVER pode-se obter os parâmetros específicos para enviar e receber os dados desejados. (A seção 3.6.1 mostra o programa TOPSERVER e como utilizá-lo)

Tabela 3.2: Outras variáveis utilizadas no projeto.

Nome	Classificação
SD1_D-Desliga saída Digital 1	Interna
SD2_D-Desliga saída Digital 2	Interna
SD3_D-Desliga saída Digital 3	Interna
SD4_D-Desliga saída Digital 4	Interna
SD5_D-Desliga saída Digital 5	Interna
SD6_D-Desliga saída Digital 6	Interna
SD7_D-Desliga saída Digital 7	Interna
SD8_D-Desliga saída Digital 8	Interna
SD1_L-Liga saída Digital 1	Interna
SD2_L-Liga saída Digital 2	Interna
SD3-Liga saída Digital 3	Interna
SD4_L-Liga saída Digital 4	Interna
SD5_L-Liga saída Digital 5	Interna
SD6_L-Liga saída Digital 6	Interna
SD7_L-Liga saída Digital 7	Interna
SD8_L-Liga saída Digital 8	Interna
POS_X-posição em X	Interna
POS_Y-posição em Y	Interna
AMB1-Ambiente 1	Interna
AMB2-Ambiente 2	Interna
AMB3-Ambiente 3	Interna
AMB4-Ambiente 4	Interna
AMB5-Ambiente 5	Interna
AMB6-Ambiente 6	Interna

3.2.2 Configurações na tela

Uma parte importante do trabalho é a amostragem das variáveis necessárias e a animação que se deseja. No projeto foi implementado um sistema de localização no laboratório LARA, sendo necessário mostrar a pessoa localizada na tela se movimentando a cada vez que o valor fosse atualizado pelo sistema de localização (seção 3.4) fazendo assim necessário um script que cumprisse tal objetivo. Para termos maior visibilidade foi criado uma amostragem da posição da pessoa e também amostradores do estado em que cada aparelho de ar condicionado se encontra.

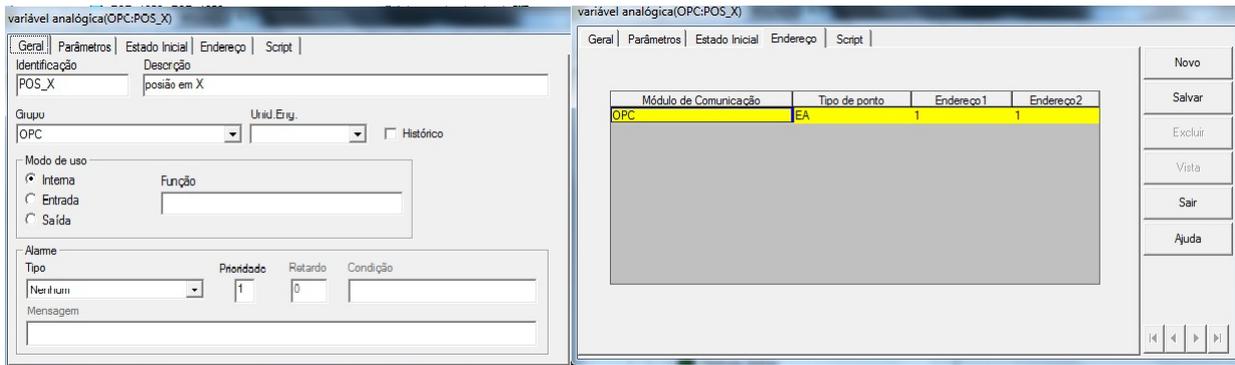


Figura 3.1: Configuração dos parâmetros da variável analógica OPC :POS_X.

3.2.3 Estado do ar condicionado

Para que o estado do ar condicionado possa variar (ligado ou desligado), é necessário se configurar a variável conforme a Figura 3.2, as configurações mostradas foram as utilizadas no presente projeto. As figuras utilizadas são produzidas externamente e precisam ser colocadas na pasta figuras dentro do arquivo do projeto. Não esquecer de mudar a quantidade de estados para 2 e também a rotina CmdDJ1. É preciso configurar o grupo da variável, dar um nome para a variável e selecionar qual figura irá aparecer em cada estado dessa variável (figura 3.2).

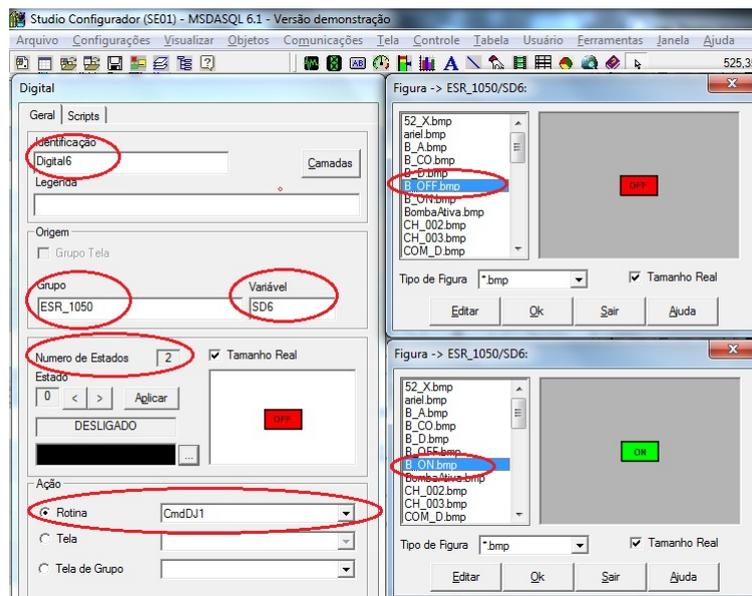


Figura 3.2: Configuração dos parâmetros da variável de estado que aciona o aparelho de ar condicionado.

3.2.4 Script

Uma maneira que foi encontrada de se ter em tempo real a localização de uma determinada pessoa dentro do ambiente, foi fazer isso através do script fornecido ao ActionView. Para que

possamos escrever o script é preciso primeiramente clicar duas vezes na tela principal do ActionView e aparecerá uma tela auxiliar, na qual deve-se escolher a opção on timer, conforme o anexoB mostra.

Uma tela para programação do *script* surgirá, onde no código implementado, foram utilizadas algumas variáveis internas declaradas como :

```
DIM VAR1_X
```

A aspa simples é utilizada para comentar uma linha do código. Para pegarmos o valor da posição onde se encontra determinada pessoa (calculada pela rede neural no Matlab) através do OPC, necessita-se do comando:

```
Var.Value("VARIABLES_ACESSO", "POS1_Y") = Var.Value("OPC","POS_Y")
```

`Var.Value("VARIABLES_ACESSO", "POS1_X") = Var.Value("OPC","POS_X")` onde `Var.Value` possui duas variáveis, sendo a primeira o grupo e o segundo parâmetro é o nome da variável que se deseja passar o valor. Os valores que são fornecidos pela rede neural, precisam passar por um ajuste devido ao tamanho da tela no computador, para tal, foi feito um comando de ajuste dos valores recebidos através de uma função linear equacionada através dos valores originais em relação aos valores da tela, conforme descrito abaixo :

```
Controls.Item("LARA|Digital0").Top = (-Var.Value("VARIABLES_ACESSO", "POS1_Y")  
+ 11)*1050.090909
```

```
Controls.Item("LARA|Digital0").Left = (-Var.Value("VARIABLES_ACESSO", "POS1_X")  
+ 18)*1000.5
```

Para que pudesse ser definido uma área na tela correspondente a cada ambiente foi delimitado por operações lógicas and e o menor que e maior que através do, como podemos ver nas linhas do script abaixo:

- Ambiente 1

```
if Var.Value("VARIABLES_ACESSO", "POS1_X")< 5 and Var.Value("VARIABLES_ACESSO",  
"POS1_X") > 2 and Var.Value("VARIABLES_ACESSO", "POS1_Y")< 5 then
```

Primeiramente foi pensado no caso específico para uma pessoa portando a Tag ativa, o que havia sido já implementado por trabalhos anteriores, culminando em uma lógica simples para acionamento dos aparelhos de ar condicionados. Se a pessoa se encontra dentro do ambiente, é incrementado de 1 a cada segundo, para acionar conforme um tempo mínimo de permanência no mesmo, conforme as linhas de código abaixo mostram para o ambiente 1.

```
Var.Value("VARIABLES_ACESSO", "AMB2")=0
```

```
Var.Value("VARIABLES_ACESSO", "AMB3")=0
```

```
Var.Value("VARIABLES_ACESSO", "AMB4")=0
```

```
Var.Value("VARIABLES_ACESSO", "AMB5")=0
```

```
Var.Value("VARIABLES_ACESSO", "AMB6")=0
```

```

Var.Value("VARIABLES_ACESSO", "AMB1")= Var.Value("VARIABLES_ACESSO", "AMB1")+1
'CONTADOR PARA LGAR O ARCONDICIONADO 1 DEPOIS DE X SEGUNDOS, NO
CASO X=6
Var.Value("VARIABLES_ACESSO", "AMBIENTE")= Var.Value("VARIABLES_ACESSO",
"AMB1")
if Var.Value("VARIABLES_ACESSO", "AMB1")=6 then
Call Command("ESR_1050","SD1_L",6,,,Var.CommandParameter("ESR_1050","SD1_L"),false)
end if
else
Call Command("ESR_1050","SD1_D",6,,,Var.CommandParameter("ESR_1050","SD1_D"),false)

```

3.3 RFID

Radio frequency identification (RFID) é um termo genérico que é usado para descrever um sistema que transmite a identificação (na forma de um número serial único) de um objeto ou pessoa utilizando ondas de rádio.

Uma *tag RFID* consiste de um microchip ligado a uma antena de rádio. O *chip* da *WAVE-TREND* pode armazenar 2 Kbytes de dados. Para receber os dados armazenados na *tag*, faz-se necessária uma leitora. Uma leitora típica é uma antena receptora de dados que podem ser acessados com o devido código, como é mostrado no Anexo D.

Cada receptor com sua antena tem um alcance de 30 m (a especificação do fabricante é que esse alcance seria de 30 m, mas na prática só obtivemos medidas até 7 m devido as interferências no ambiente e aos multicaminhos) . A área de cobertura teórica da antena é omnidirecional com radiância espectral circular. Assim, tendo em vista a desejada redundância de medidas, 5 receptores *Wavetrend RX1000* foram adquiridos e instalados em locais espaçados do laboratório. O equipamento utilizado é mostrado na Figura (3.3)

A *tag* ativa escolhida para o projeto possui bateria própria e envia seu sinal de tempos em tempos, podendo esse intervalo ser programável pelo fabricante, se assim se solicitar, de acordo com a tabela fornecida, mostrada no Anexo F. A localização RFID utiliza a potência do sinal recebido para estimar a distância da *tag* ao módulo fixo, no qual a potência padrão de espaço livre de sinais é dada pela Eq.3.1 em dB e chamado de valor RSSI da *tag*, que estima a distância ao receptor por intermédio desse valor de sinal, que é mandado pela *tag* através do pacote de dados nos Bytes 26-29, conforme descrito no Anexo F. onde P é a potência, k uma constante do meio de propagação e D é distância da *tag* ao receptor.

$$p = \frac{k}{D^2}. \quad (3.1)$$

O que seria algo simples na teoria, na prática se torna algo muito complicado, pois o comporta-



Figura 3.3: Foto da leitora *Wavetrend*, antena e *tag* ativa [9].

mento da potência das ondas de rádio não se dá de forma tão simples como descrito por Eq.(3.1), pois temos muita influência do ambiente em que as mesmas estão se propagando, como: reflexões, refrações, ruídos e outras interferências que modificam a relação distância versus potência, fazendo com que o RSSI não seja constante para distâncias sobre o raio D .

Feitos alguns testes no LARA, foi notado que para locais diferentes uma mesma tag possui o mesmo RSSI para um mesmo receptor, o que se deve aos multi-caminhos e atenuação, ou seja, um único sinal pode chegar ao receptor por vários caminhos diferentes dependendo das pessoas no ambiente, dos objetos e outros fatores, resultando assim em vários sinais diferentes para uma mesma posição com relação a uma leitora específica. Devido a essas observações, foi proposto que se instalasse 5 leitoras no laboratório, o que não elimina o problema de multi-caminhos, mas permite melhorar a estimativa da localização.

3.4 Localização

A localização em um ambiente *indoor*, pode ser feita de inúmeras maneiras, mas neste trabalho foi feita através de uma rede neural artificial, que foi treinada para localizar uma tag de acordo com os valores RSSI fornecidos pelas leitoras. Inicialmente apresentamos o treino com *tags* estáticas e depois a utilização de robôs móveis no treino.

3.4.1 RNA-Rede Neural Artificial para *tag* estática[9]

Uma rede neural artificial é uma implementação artificial que tenta imitar a forma de processamento da rede neural humana. O paradigma da Inteligência artificial (RNAs) tenta criar formas de emular num computador o comportamento do cérebro humano, classificando e aprendendo padrões.

Uma RNA pode ser conforme ilustrado na Figura 3.4, contendo uma camada de entrada, onde se encontram as variáveis do processo, uma camada oculta (contendo os neurônios artificiais) e uma

camada de saída. Os neurônios artificiais, chamados Perceptrons, consistem de um conjunto de entradas que serão ponderadas, passando por uma função de ativação gerando uma saída. O que uma rede neural faz é ajustar os pesos de cada entrada para produzir uma saída desejada, sendo capaz de interpolar para dados apresentados à rede neural.

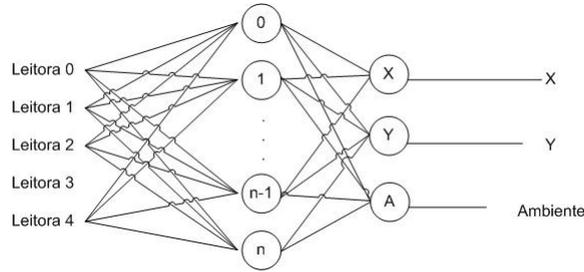


Figura 3.4: Ilustração de uma rede neural[9].

Para treinar a rede neural, são necessários o conjunto de entradas e saídas (*Pattern, Target*), diversos estados das variáveis do processo, sendo utilizado a rede MLP-*Multiplayer Perceptron* e o algoritmo de treinamento *backpropagation*. A rede MLP possui no mínimo 3 camadas: entrada (sem neurônios), camada oculta (possui neurônios), saída (possui neurônios). O algoritmo *backpropagation* consiste de duas etapas: na primeira, inicializam-se os pesos da rede de forma arbitrária e calcula-se a saída da rede. Esta saída é então comparada com a desejada e calcula-se o erro como sendo a diferença entre a saída da rede e a saída desejada. Na segunda etapa, a partir desse erro calculado, ajustam-se os pesos da camada de saída para anular esse erro, depois os pesos da camada oculta, isto é, os pesos da rede são modificados de trás pra frente. A rede MLP utilizando-se dos valores RSSI fornecidos pelas 5 leitoras dispostas no LARA conforme é mostrado na Figura 3.5, onde uma tag foi posicionada em 59 localidades dentro do laboratório e foram registrados os valores RSSI de cada receptor para cada posição. O *pattern* (P) da rede são os valores RSSI e o *target* (T) as posições selecionadas para a tag [9].

Treinando a RNA para aprender essas 59 posições do laboratório (Figura 3.6), tendo como entradas os valores RSSI dessas posições, a rede é capaz de localizar uma tag no laboratório usando a RNA. De forma sucinta: uma pessoa com uma tag numa determinada posição no laboratório irá enviar sinais para os receptores. Estes enviarão os valores RSSI relativos a essa leitura ao computador hospedeiro da RNA. Quando a rede neural recebe esses valores ela interpola com os valores por ela conhecidos e gera uma saída interpolada daquelas calibradas. Essa saída corresponde à posição da tag (da pessoa) no laboratório. No caso foi considerado apenas uma parte do LARA, devido à falta de leitura quando se distanciava mais que 7 metros das leitoras, limitando a localização à área preenchida pelos pontos amarelos na Figura (3.6).

A rede neural com 6 neurônios na camada oculta foi capaz de localizar a tag no laboratório com precisão melhor do que 1,5m. Isso é um resultado aceitável para muitas aplicações, inclusive a proposta por este trabalho. O problema surge quando a tag está próxima da fronteira entre dois ambientes onde há uma parede ou divisória. Por causa da precisão de 1,5m, um usuário sentado do lado de fora da sala de reuniões, mas próximo da divisória, poderia ser facilmente localizado do lado de dentro da sala. Isso faria com que o ar condicionado da sala fosse acionado, resultando em

3. Entretanto, foi verificado que com apenas as 6 medidas de cada ponto de treinamento que se possuía não obteve-se uma rede neural robusta o suficiente para as aplicações desejadas como pode ser observado no capítulo 4.2, logo foi modificada a maneira de se coletar os dados, ao invés de se fazer a trajetória pré-definida toda vez que se desejasse dados novos, foram coletados 25 dados em um mesmo lugar, porém com a tag transmitindo a cada 1,5 segundos, para simular o andar do robô, e depois a rede neural foi treinada, na seção 4.2 será mostrado a precisão alcançada para essa rede neural. Foi usada a rede neural perceptron multicamadas, 15 neurônios na camada oculta, 5 neurônios de entrada e 5 neurônios de saída e função de ativação tansig, mostrada na seção 2.3.1.1.

3.5 OPC

Para comunicação do sistema de localização e o ActionView, fez-se necessária a construção de uma interface OPC dos dados de localização fornecidos, onde OPC (*OLE for Process Control*) é uma interface padronizada, aberta, definida pela *Microsoft* para intercomunicar microcomputadores e IED's (Intelligent Electronic Device), utilizados em Automação Predial. Atualmente esse protocolo é mantido através de uma fundação de usuários OPC: <http://www.opcfoundation.org/>.

O módulo AVOPC.DLL trabalha, basicamente, com dados do tipo:

- Entradas analógicas: variáveis analógicas lidas dos IEDs;
- Entradas digitais: variáveis digitais lidas dos IEDs;
- Tag de tempo: tempo em que uma variável foi lida;
- saídas digitais: saídas do tipo digital dos IEDs;
- saídas analógicas: saídas do tipo analógico dos IEDs;
- Cadeias de caracteres: textos associados a variáveis dos IEDs.

3.5.1 Configuração no ActionView de Parâmetros para OPC

O canal OPC é do tipo *DUMMY* e sua ficha de parâmetros é mostrada na figura 3.7

- *Nome do Nó de rede Remoto (OPCRemoteNode)*

Nome ou IP na rede do computador remoto em que está instalado o servidor OPC que se deseja conectar. Se o servidor OPC estiver no próprio computador, pode-se deixar em branco este campo.

- *Nome do Servidor OPC (OPCServerName)*

Nome do programa Servidor OPC ao qual se deseja conectar (para se conectar ao OPC Server de outro computador, os dois devem estar no mesmo domínio e com mesmo usuário registrado).

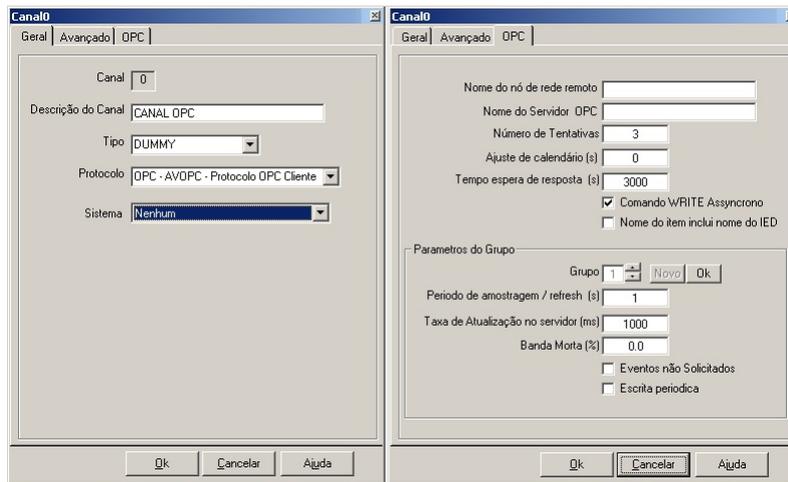


Figura 3.7: Configuração dos parâmetros para OPC do ActionView.

- *Numero de tentativas (MaxRetry = 2)*

Número de tentativas de reenvio de uma mesma mensagem de leitura para o servidor OPC. Após estas tentativas, será gerado evento de Falha de Comunicação com o servidor.

- *Escrita Periódica*

Marcando esta opção, o sistema considera que o grupo é para escrita e não para leitura. Serão executados ciclicamente, com a periodicidade definida em Período de Amostragem, acima, solicitação de escrita dos valores atuais destes tags nos itens OPC, definidos para este grupo.

Como os sistemas operacionais Windows suportam automaticamente o protocolo OPC, ele tem sido usado como "gateway" entre protocolos de IEDs, utilizados com supervisórios que executam em ambiente Windows.

Todos os servidores OPC permitem, através do uso de um browser, a criação de:

- Canais de comunicação;
- IEDs existentes em um canal;
- Pontos de entrada e saída, analógicos e digitais existentes em nós IEDs

Para se conhecer os pontos existentes em um IED conectado por um Servidor OPC, pode-se conectar ao servidor um cliente OPC com *browser*. Assim, se conhecerá os pontos existentes e sua tipificação em OPC. No cadastramento de pontos na tabela CANAIS, há uma facilidade para mostrar os pontos dos servidores e fazer a transferência de ItensID diretamente para a tabela canais.

Protocolo	Grupo	Variável	Tipo	Endereco1	Endereco2	Descrição
OPC	AV_AR1	DBE1	SA	1:1	KMD5901.AR.DEADE1C1	Temp. Min. E1 - C1
OPC	AV_AR1	DBE2	SA	1:1	KMD5901.AR.DEADE2C1	Temp. Min. E2 - C1
OPC	AV_AR1	E1CX	ED	1:1	KMD5901.AR.E1C1_LD	Estágio 1 C1
OPC	AV_AR1	E2CX	ED	1:1	KMD5901.AR.E2C1_LD	Estágio 2 C1
OPC	AV_AR1	ESMA	ED	1:1	KMD5901.AR.ESMA_C1	Estágio status
OPC	AV_AR1	ESMA_D	SD	1:1	KMD5901.AR.ESMA_C1	Off Manual Estágio
OPC	AV_AR1	ESMA_L	SD	1:1	KMD5901.AR.ESMA_C1	On Manual Estágio
OPC	AV_AR1	EVA	ED	2:2	KMD5902.HI.LD-EVACT	Fancoil Status
OPC	AV_AR1	EVA_D	SD	2:2	KMD5902.HI.LD-EVACT	Off Fancoil
OPC	AV_AR1	EVA_L	SD	2:2	KMD5902.HI.LD-EVACT	On Fancoil
OPC	AV_AR1	EVAPQ	ED	2:2	KMD5902.HI.UMI_1	Evaporator Status

Figura 3.8: Endereçamento dos pontos na tabela CANAISPEC.

3.5.2 Endereçamento dos pontos na tabela CANAISPEC

O endereçamento físico na tabela CANAISPEC é o seguinte: **Endereço1** - Tem a forma de um par de números separados por ':' como em <endereço_device>:<grupo> onde:

- <endereço_device> É um endereço físico do IED, único entre todos os IEDs conectadas a este servidor de 1 a 32767.
- <grupo> é o número de um grupo de pontos.

Endereço2 - É o texto do ItemID deste item para o servidor OPC. Há duas formas de especificação deste campo, dependendo de como for feita a configuração de parâmetros do canal (parâmetro: Nome do item inclui nome do IED).

- Especificar apenas a parte que não é usada para definir o device conectado no servidor. O módulo de comunicação OPC, ao criar o item no servidor, juntará o DeviceName, com esta parte do ItemID para formar o ItemID completo. (Esta junção é apenas a concatenação dos dois strings. Se houver necessidade de um separador, como uma '/' ou um ponto, o mesmo deve ser incluído na definição do nome do IED).

3.6 Aquisição de dados das leitoras no Matlab

O recebimento dos dados fornecidos pelas leitoras precisam ser disponibilizados a quem quer que precise deles, o que não é possível quando tem algum usuário realizando leitura, pensando nisso e nos fins de implementação real do projeto, foi feito um programa em Matlab que lê esses dados das leitoras e os disponibiliza para o supervisor ActionView via OPC.

Foi desenvolvido um script no programa MatLab para a aquisição dos dados da *tag*, para uma precisão maior no tempo de amostragem dos RSSIs e também devido ao fato de se precisar passar os dados para a rede neural e depois para o software SCADA ActionView via OPC o que seria inviável com o programa em Delphi disponível no laboratório, pois o mesmo não fornecia os dados para outros usuários. O primeiro script feito foi para armazenar os dados em um arquivo txt para posterior análise dos dados e treinamento da rede neural, que é mostrado no diagrama de blocos da figura 3.9.

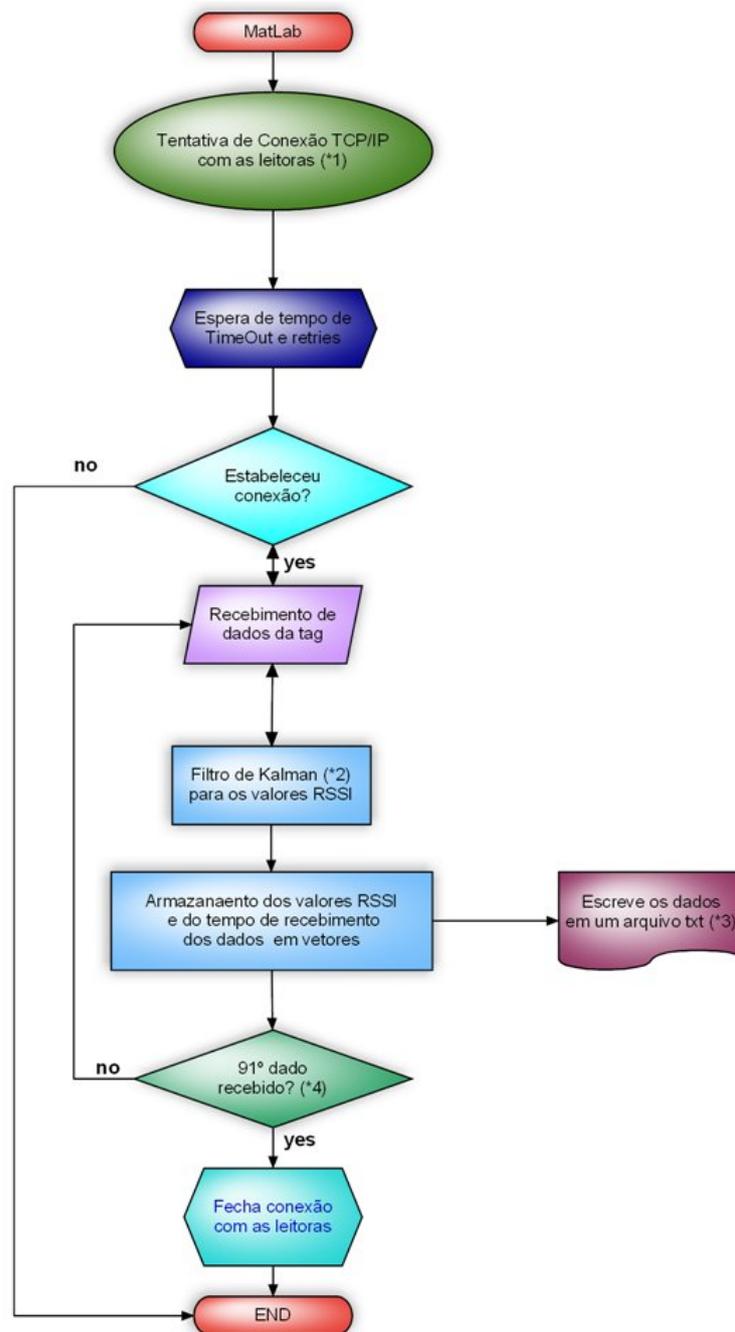


Figura 3.9: Diagrama de blocos para o programa em Matlab de aquisição de dados das Leitoras.

3.6.1 OPC em MatLab

Para fins de projeto profissional, existe uma grande necessidade do uso de softwares mais específicos. Com isso em mente, foi proposto que os dados sejam processados no Matlab com suporte OPC.

O Matlab oferece algumas funções de cliente OPC, que foram implementadas na comunicação dos dados. O código do programa em Matlab se encontra no anexo D. Para inicializar uma comunicação OPC temos de criar um cliente do tipo *Data Aquisition*, podendo ser conectado

internamente através do parâmetro de comunicação, 'localhost', ou externamente através do IP da máquina que se deseja comunicar, referenciando também o nome do servidor, mandando o comando para conectar conforme o exemplo abaixo:

```
dadoTag = opcda('localhost','SWToolbox.TOPServer');
connect (dadoTag);
```

Posteriormente adiciona-se os grupos do servidor, para poder ter acesso às chamadas *tags*, que podem ser equiparados a *end Device*, através da função *additem*, sendo o parâmetro conforme o que o servidor fornece, não podendo esquecer o tipo do dado (*Double*, *Single dentre outros*), conforme as linhas do código abaixo:

```
Leitora_01 = addgroup (dadoTag, 'Group1' );
Leitora_01_Tag_01 = additem (Leitora_01,'Channel1.Device1.Group1.tag1','double');
```

Depois que o 'caminho' foi definido, se a variável do OPC é do tipo *read and write*, o cliente poderá ler e escrever no OPC. a função no Matlab que lê está mostrado abaixo:

```
Val_01_Tag_01= read(Leitora_01);
[ RSSI_01_01 RSSI_01_02]= Val_01_Tag_01.Value;
```

Para validar a comunicação via OPC foi instalado um simulador de servidor OPC chamado *TOPSERVER*, necessitando fazer alguns ajustes de parâmetros iniciais do simulador de servidor OPC.

Para melhor visualização, a Figura 3.10 mostra a tela principal do simulador de servidor Top-Server.

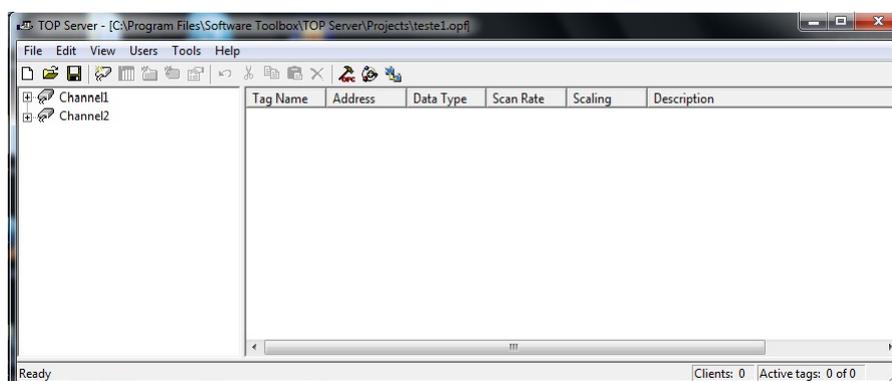


Figura 3.10: Tela principal do Simulador de Servidor OPC.

Após colocar os parâmetros a configuração do TopServer fica aproximadamente como mostra a Figura 3.11.

Para que a conexão com o OPC do ActionView com o Matlab fosse feita, foi necessária a criação de algumas linhas de código específicas, com os nomes das variáveis no ActionView, que foram vistos pelo simulador de cliente OPC mostrado anteriormente seguindo o diagrama de blocos mostrado na figura 3.12. O código completo pode ser visto no Anexo D.

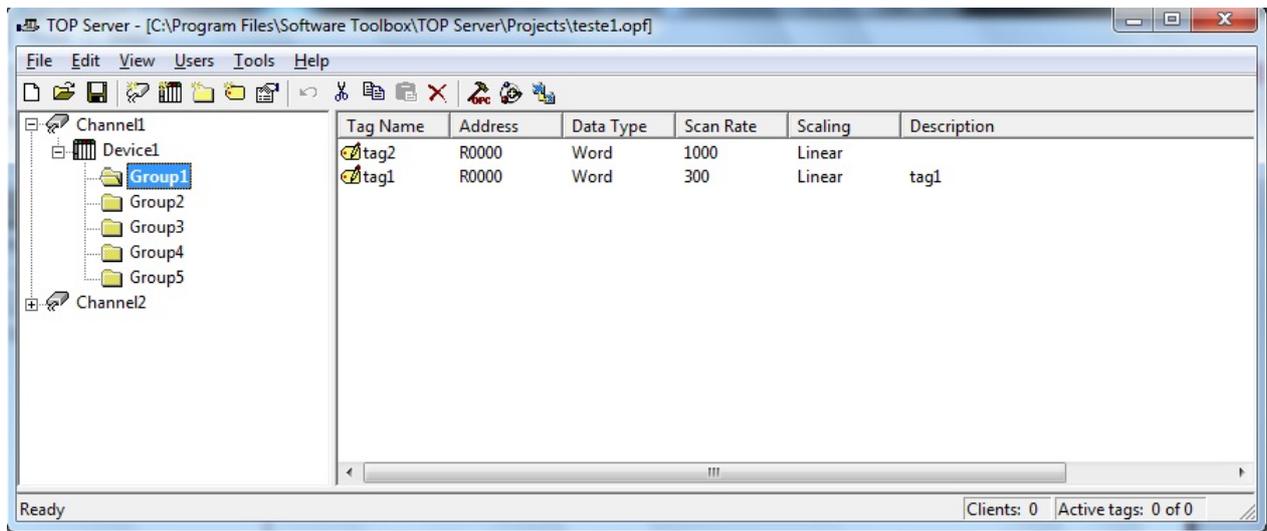


Figura 3.11: Configuração dos parâmetros do TopServer.

3.7 MODBUS

Optou-se por utilizar a comunicação via Modbus. O ActionView já possui o protocolo Modbus. Para melhor configuração e projetos futuros são citados alguns parâmetros e modos de uso do Modbus no ActionView.

O módulo AVMODBUS.DLL disponibiliza o protocolo MODBUS mestre o escravo em três opções: Modbus-RTU, Modbus-ASCII e Modbus-TCP/IP.

No modo mestre, o módulo executa a aquisição de dados através da solicitação de leituras e escritas para equipamentos escravo. No modo escravo, trabalha de forma passiva, recebendo solicitações de leitura ou escrita de algum outro equipamento ou supervisor, respondendo com os estados e valores atualmente existentes na base de dados em tempo real do ActionView. No caso de recebimento de solicitações de escrita, executa estes comandos sobre o ActionView local. Neste modo, pode-se montar configurações de "gateways" entre o protocolo Modbus e outros protocolos mestre disponíveis no ActionView. De forma a permitir na parametrização, que se defina o tipo de função de leitura e a forma da representação dos dados, foram definidos vários tipos de pontos que permitem tais escolhas, conforme o formato da base de dados do equipamento sendo utilizado como na figura 3.13, [3].

A/D - A = ActionView trata como analógico. D=ActionView trata como Digital. I / O - I = ponto de entrada; O = Ponto de saída (comando ou escrita).

No programa feito para a implementação do projeto foram utilizados I/Os do tipo OR-*holding registers*, IR- *input registers* e output registers, pois foi o que o equipamento utilizado no projeto exigiu, um relé (ESR-1050), disponibilizado pela empresa Spin Engenharia de automação. Imagens do mesmo podem ser vistas na Figura 3.14, com entradas e saídas disponíveis para acionamento, porém para os fins do projeto ainda se necessita de um módulo de potência (Seção 3.8) na saída devido à tensão que os aparelhos de ar-condicionados exigem.

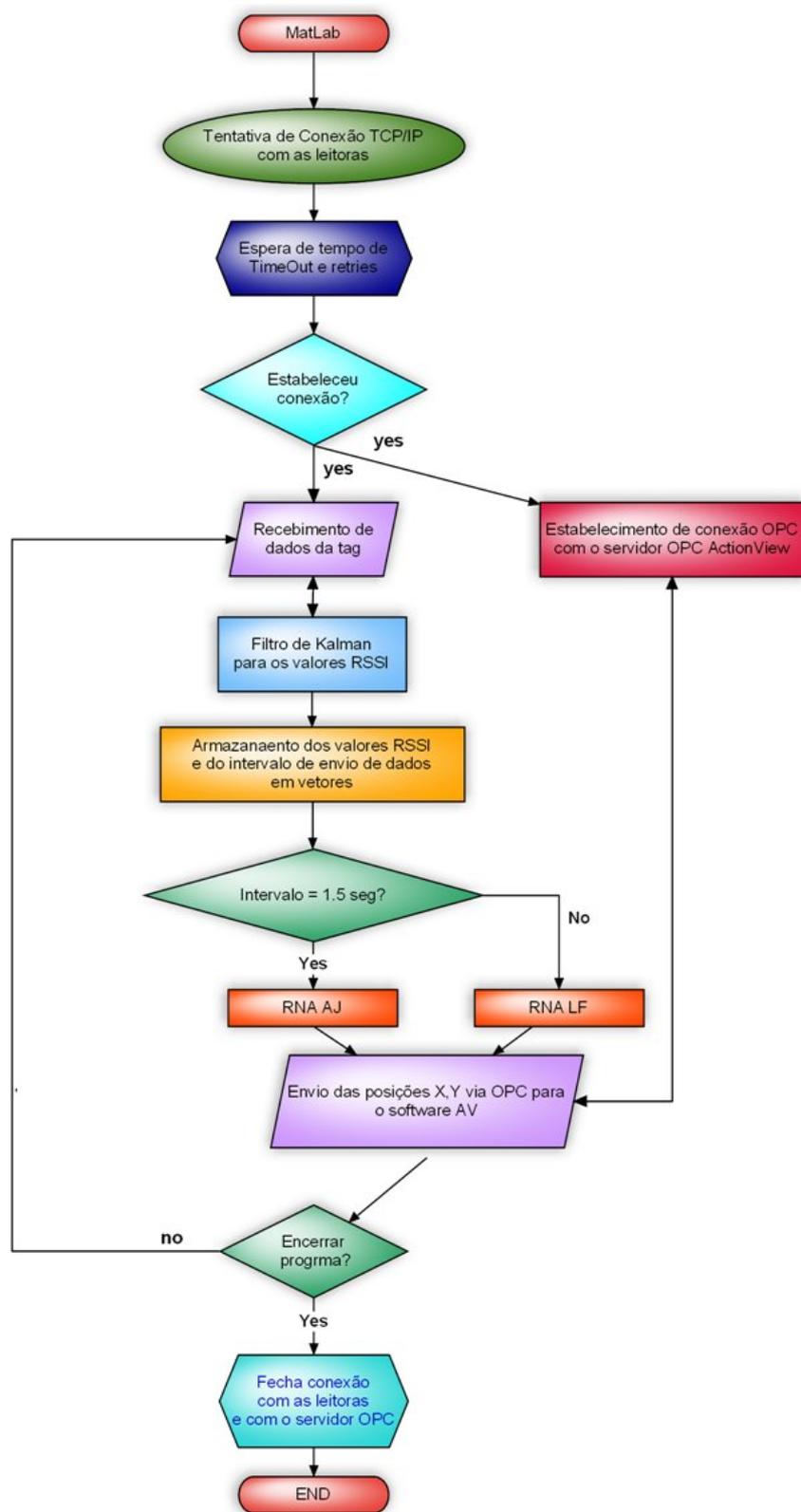


Figura 3.12: Diagrama de blocos do programa em Matlab para comunicação via OPC.

Para maiores e melhores informações a respeito da implementação Modbus faz-se referência ao Manual ActionView Protocolos.

Num	Sigla	A/D	Tipo	Descrição / Função Utilizada	Modulo
0	OS	D	I	Output Status – Bits	MODBUS
1	IS	D	I	Input Status – Bits	MODBUS
3	IR	A	I	Input Register – 16 bit Word	MODBUS
4	OR	A	I	Output Register – 16 bit Word	MODBUS
6	SY	D	I	Pontos do Sistema (comunicação)	MODBUS
13	SIR	A	I	Input Register - Signed	MODBUS
14	SOR	A	I	Output Register - Signed	MODBUS
23	FIR	A	I	Input Register – Float (2 Words)	MODBUS
24	FOR	A	I	Output Register – Float (2Words)	MODBUS
43	DIR	A	I	Input Register – double (4 Words)	MODBUS
44	DOR	A	I	Output Register – double (4 Words)	MODBUS
53	LIR	A	I	Input Register – Long (Inteiro 32 bits)	MODBUS
54	LOR	A	I	Output Register – Long (Inteiro 32 Bits)	MODBUS
63	BIR	D	I	Input Register – Acesso a Bits	MODBUS
64	BOR	D	I	Output Register – Acesso a Bits	MODBUS
7	SD	D	O	Preset Single Register (saida digital em register de 16 bits)	MODBUS
9	SA	A	O	Preset Single Register (saida analógica, inteiro 16 bits)	MODBUS
5	EV	D	I	Tratamento Especial de Eventos COM alteração do estado na base de dados	MODBUS
8	EVX	D	I	Tratamento Especial de Eventos SEM alteração do estado na base de dados	MODBUS
65	SFC	D	O	Force Single Coil – saída digital em um bit, conforme parâmetro	MODBUS

Figura 3.13: Tipos de ponto no Modbus.



Figura 3.14: Gravuras do Módulo de comunicação Modbus e suas conexões.

Para que o sinal enviado pelo supervisor chegue até o relé necessitava-se de algumas etapas de comunicação, também disponibilizado pela SPIN Engenharia. Os dois fios que saem da placa são os que enviam e recebem os dados da ESR-1050, sendo os mesmos ligados na entrada D+ e D- do protótipo. O *Smart Converter* necessita de uma fonte externa para acionar seus circuitos internos e possui uma entrada RS 422/485 e outra RS 232 a que foi a utilizada como conexão serial com o PC supervisor.

Como o projeto está sendo realizado em computadores que não possuem uma saída serial, foi necessário se utilizar um conversor serial-USB(Figura 3.14). A instalação do conversor foi feita conforme o manual *AN 119 FTDI Drivers Installation Guide for Windows7*.

3.8 Módulos de potência

O acionamento dos aparelhos de ar-condicionados não pode ser feito apenas pela saída do relé ESR-1050, pois o mesmo fornece uma saída apenas uma voltagem de 3V, sendo assim necessário amplificar a tensão e a corrente projetando um módulo de potência que consiga fornecer a tensão e a corrente exigidas para o acionamento dos aparelhos de ar-condicionado.

Foram estudados alguns parâmetros do equipamento para se ter idéia do que seria necessário para a implementação do módulo e verificou-se que um relé de 240V e 1 A seria suficiente para o ar-condicionado da sala de reunião e também os ar-condicionados *split*.

3.9 Integração final

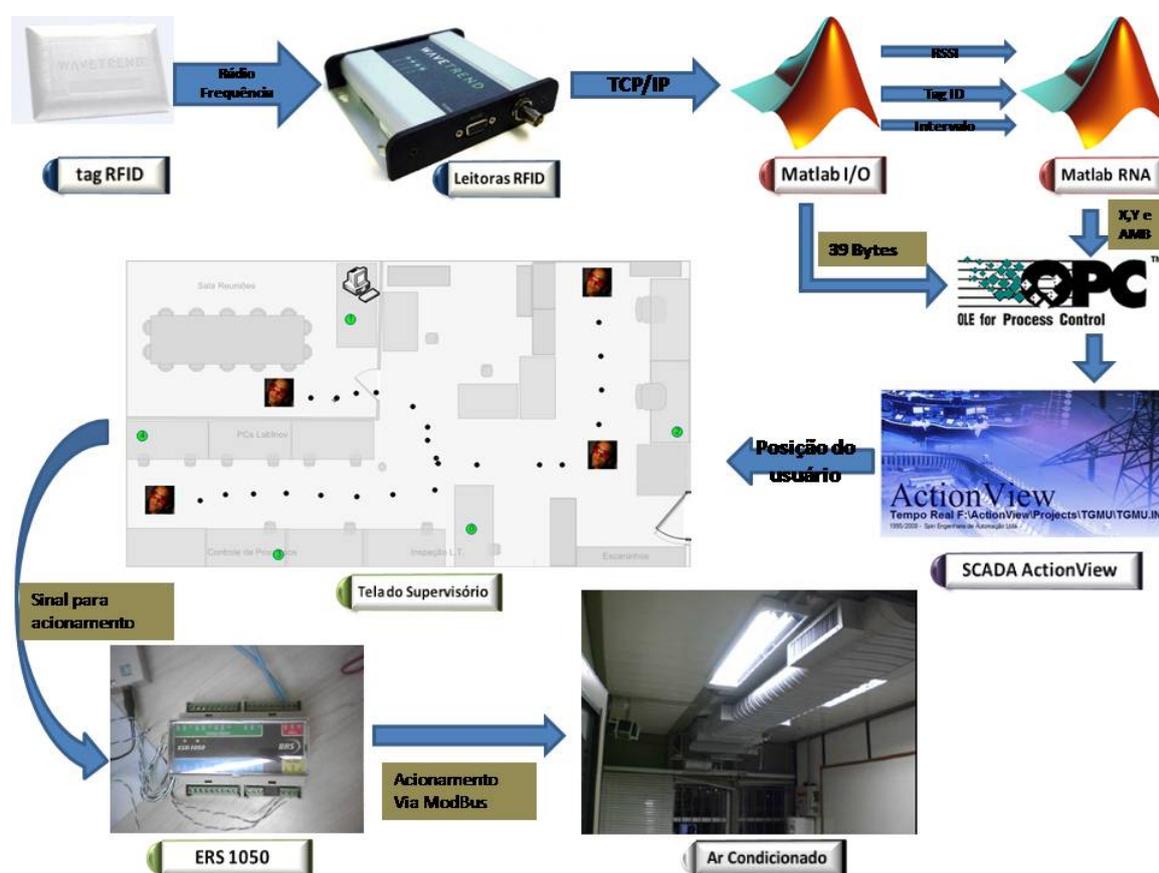


Figura 3.15: Visualização de todos os módulos do projeto (“painel mímico”).

Com todos os passos anteriores realizados e funcionando, foi então possível integrar cada uma das partes do projeto. Os dados lidos pelo Matlab são enviados para o ActionView, que é o nosso Servidor OPC, fornecendo dados para clientes OPC, não mais limitando o uso dos dados ao cliente que fez a conexão inicial com a leitora, permitindo que trabalhos sejam feitos de forma paralela no LARA conforme a figura 3.15 mostra.

As informações que são enviadas por cada *tag*, contém 39 bytes(anexo F), possuindo o Header,

Da borneira (saída digital do relé ESR 1050) é que se tem a saída para o acionamento dos aparelhos de ar-condicionados. O módulo utilizado envia o sinal de forma inversa, pois quando a saída é ativada fornece 0V e quando está desligada fornece 3V, o que tornou necessário um inversor na sua saída. Havia algumas maneiras de se fazer essa inversão, pois a corrente fornecida pelo módulo não é suficiente para acionar o relé, por isso foi utilizado uma porta NOT (TL 074), que é alimentada com 0V no *ground* e 5V no *VCC*, essa alimentação fornece uma corrente para a saída do CI, sendo o suficiente para acionar o relé que é normalmente aberto.

Com todas as funcionalidades do sistema operando de forma correta, foi feita a conexão do relé com o sistema que aciona o aparelho de ar-condicionado. Como a localidade que foi definida para ser acionada possui 3 aparelhos de ar-condicionados foram utilizados 3 circuitos de potência e chaveamento, com transistor e relé (Figura 3.17), onde a entrada do relé vem do aparelho, e a saída o conecta ao *Terra*. Nos aparelhos Split a conexão não foi realizada. O Ar-condicionado da Sala de reunião é do tipo Híbrido.[8]



Figura 3.17: Relé T2405Z-M utilizado para acionamento dos aparelhos ar condicionado (Tectronic).

O sistema híbrido, figura 3.18 é composto por um sistema evaporativo e por um sistema de refrigeração convencional, que procura combinar as vantagens de ambos os sistemas com a finalidade de consumir menos energia quando comparado a sistemas convencionais e fornece também ar refrigerado de melhor qualidade (renovação do ar).



Figura 3.18: Ar condicionado híbrido utilizado no projeto.

Com todos esses passos realizados, é preciso validar o projeto em todos os seus âmbitos.

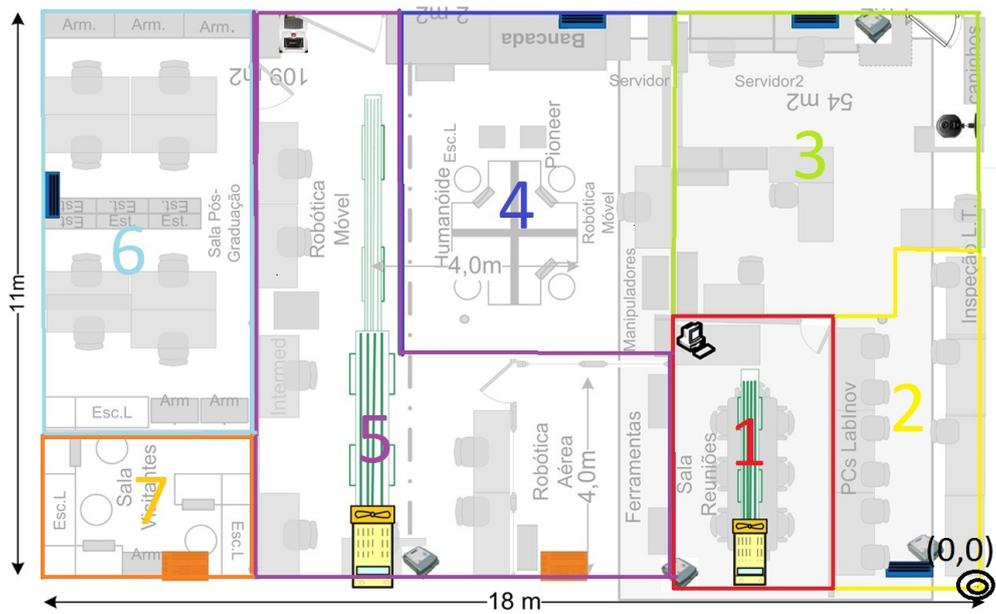


Figura 3.19: Divisão dos ambientes para acionamento setorizado do ar-condicionado.

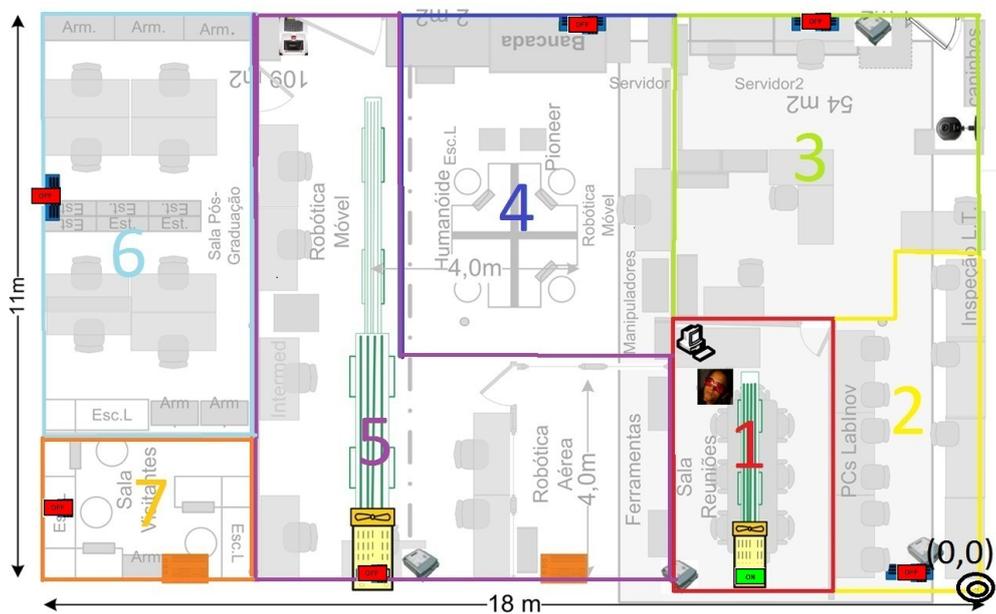


Figura 3.20: ActionView funcionando para o ambiente 1.

Capítulo 4

Resultados do projeto

Para validar o projeto, foram efetuadas alguns protocolos de teste, para a validação dos objetivos impostos.

4.1 Coleta de dados das Leitoras

A coleta de dados é essencial para o projeto, pois através dos mesmos, a rede neural foi treinada, e com esses dados pode-se verificar a localização realizada no mesmo.

Primeiramente foi pensado na utilização da *tag* utilizada em projetos anteriores no LARA [9], porém a mesma não foi encontrada no laboratório, o que levou a uma nova proposta: localizar uma pessoa andando no ambiente, já que a proposta de trabalhos anteriores foi a localização com a *tag* parada, transmitindo de 15 em 15 segundos, e em cima da mesa.

Para a primeira coleta de dados, o protocolo foi o seguinte: foram demarcados 91 passos ao longo do laboratório, conforme a figura 4.2 mostra, sendo esses pontos pré-determinados em X e Y conforme a tabela E.1 no anexo E específica. A pessoa portando uma *tag* deveria andar por esses pontos conhecidos, avançando de 1,5 em 1,5 segundos (tempo de transmissão da *tag* quando em movimento), os resultados não foram satisfatórios pois aproximadamente 20% dos dados se perderam e não chegaram para validação.

Em vista do primeiro protocolo de coleta de dados, foi proposto que a coleta dos dados fosse feita com o robô aramis do LARA 4.3, tentando assim eliminar a interferência humana do processo. Os resultados não foram satisfatórios também, pois muitos dados se perderam durante o percurso e também os dados estavam variando muito como se pode observar na figura 4.1.

4.2 Rede Neural Artificial

Em uma tentativa de treinarmos uma rede com o maior número possível de dados por ponto específico, foi proposto que a pessoa portando a *tag* ficasse movimentando a tag para que ela não deixasse de emitir a cada 1,5 segundos. Essa metodologia foi abandonada após se verificar que a interferência humana era grande e que dependendo da posição em que se encontrava, no mesmo

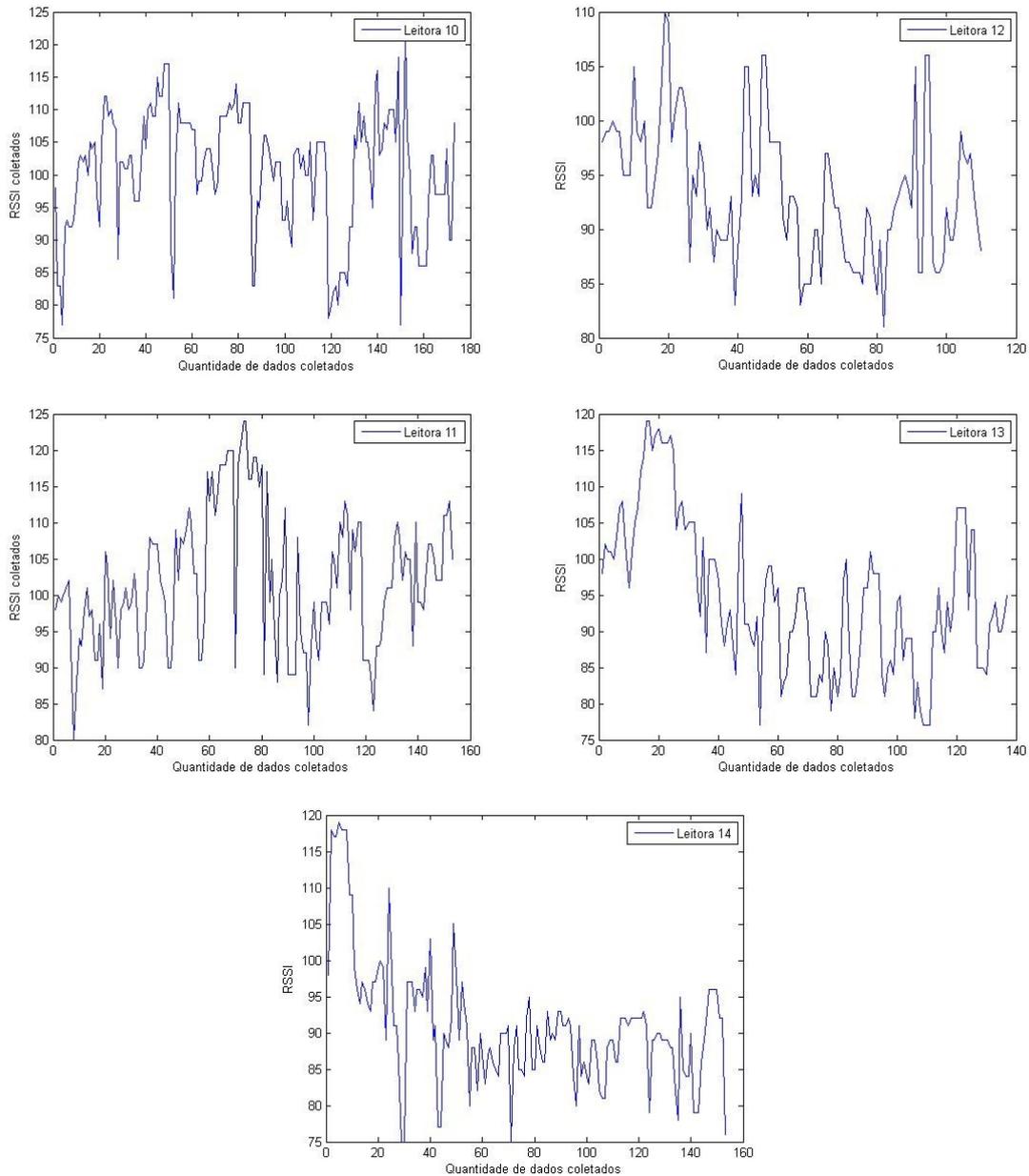


Figura 4.1: RSSI coletados, para o robô deslocando-se, aquisição de dados a 1,5 segundos.

ponto porém rotacionado de 180° , o valor RSSI diminuía bastante, chegando a variar de 20 a 30 RSSI em módulo. Com essa discrepância, o mesmo ponto, porém rotacionado, teria uma variação muito grande, impossibilitando um bom treinamento da rede neural artificial.

Uma nova proposta foi elaborada, onde um script em Matlab foi feito, passando os dados primeiramente por um filtro de Kalman [11], e somente quando o robô ARAMIS da pioneer chegasse ao ponto desejado, pressionando-se enter, o dado era gravado em um arquivo, contendo o RSSI de cada leitora, o horário em que foi coletado e a posição X,Y do mesmo. Observamos uma certa coerência nos dados para um conjunto de dados, visto que foram feitas 6 coletas de dados com o robô em movimento. Alguns dados foram coletados com o robô ARAMIS sem fio, o que diminuía a

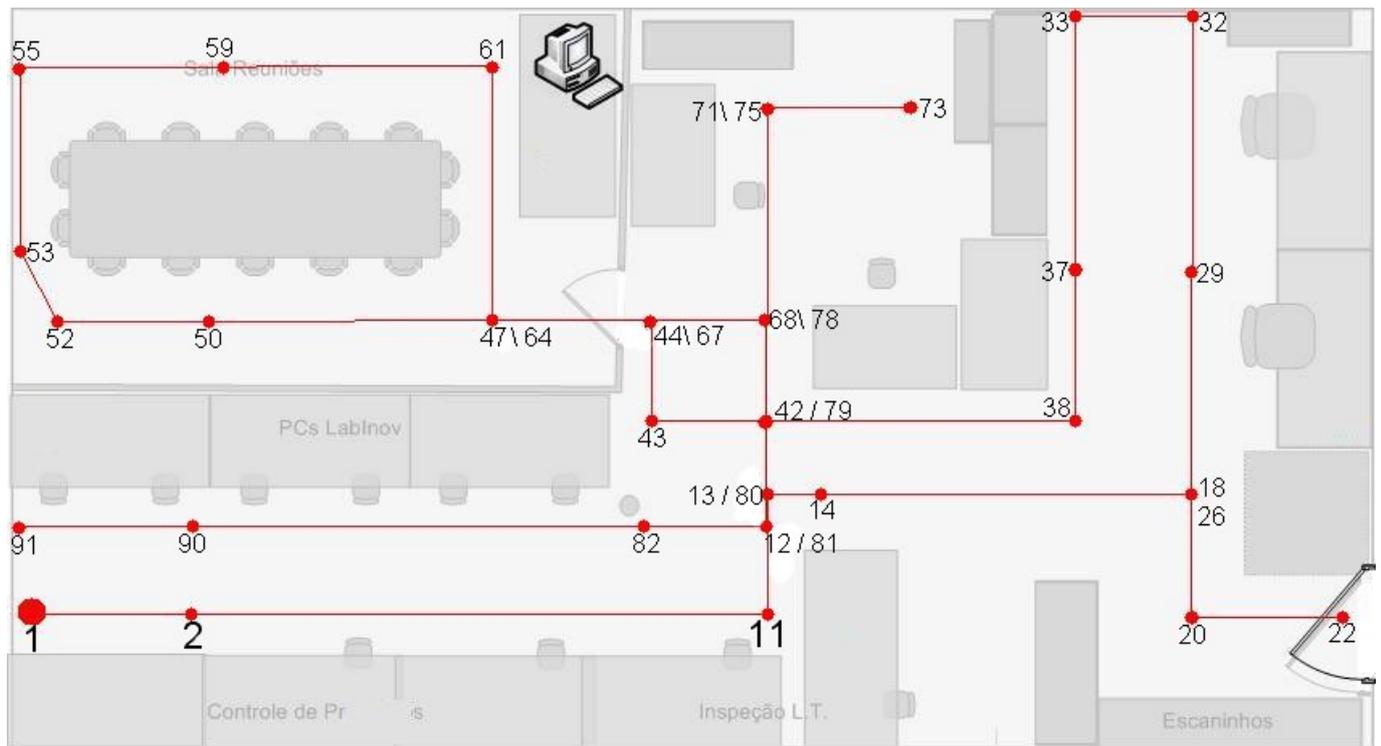


Figura 4.2: Caminho contendo os 91 pontos escolhidos.

interferência humana, pois poderia se manter uma distância razoável da *tag*, o que não era possível com o robô manipulado com o controle com fio.

Para fins de análise, a figura 4.4 foi plotado, onde os dados foram coletados sem a presença de humanos por perto e com o robô controlado por controle sem fio. A outra imagem plotada, figura 4.5 é da mesma leitora, porém com a presença de uma pessoa perto da mesma, onde a seta indica a localização da leitora 10. Comparando ambos os gráficos notamos que o RSSI é atenuado, e a superfície fica totalmente modificada de acordo com a presença de uma pessoa ou não.

Outras coletas de dados foram realizadas, sendo discrepantes também, como se pode observar nas figuras 4.6 e 4.7, todos os gráficos da leitora 10.

Pode-se ainda fazer uma comparação com os resultados obtidos para a *tag* parada sobre a mesa transmitindo a uma taxa de 15 segundos, conforme a figura 4.8, atentando para o fato de que os RSSI's estão deslocados de 70 e as posições X,Y estão em centímetros e não em metros. Podemos perceber que o valor máximo de RSSI obtido na superfície da figura 4.8, somado de 70 é de 110 [9], enquanto que o RSSI máximo para a figura 4.4 foi de aproximadamente de 135, mostrando que a taxa de transmissão dos dados influencia no valor de RSSI.

O resultado esperado é o mostrado na figura 4.9, mas quando a RNA foi testada para os dados treinados o que se obteve como saída foram os dados plotados na figura 4.10 isso por causa das discrepâncias entre os dados coletados, logo a rede neural não pode chegar ao erro zero.

Em vista dos dados e superfícies amostradas, foi proposto que o sistema possuísse duas redes neurais artificiais, uma para quando a *tag* emitisse de 15 em 15 segundos (*tag* parada) e outra para quando a *tag* emitir de 1,5 em 1,5 segundos (*tag* movimentando), já que os dados contém tal informação, foi um modo encontrado de se minimizar esse efeito, conforme o diagrama de blocos da figura 3.12 mostra.

Para a rede neural implementada como citado anteriormente, foram feitas várias medidas, na tentativa de se obter dados que fossem suficientemente bons para o treinamento de uma rede neural robusta, entretanto as abordagens feitas estavam apresentando discrepâncias devido a interferências de pessoas no ambiente de coleta de dados, à falta de repetibilidade exata das condições de coleta de dados e à complexidade apresentada pelas *tags* na hora de transmitir seus dados como pode ser visto na tabela 4.1 Logo foi proposto uma coleta de dados que colhesse 25 pacotes de dados em cada ponto, com a *tag* transmitindo a cada 1.5 segundos, esses dados foram apresentados à rede neural e a mesma foi treinada, apresentando o resultado mostrado na figura 4.11 como pode ser observado, o treinamento foi feito somente para uma parte do laboratório, o ambiente 2. A rede neural apresentou um resultado razoável para a localização, com a interpolação para os dados coletados (tabela 4.2), fato que não foi observado quando se treinou a rede neural com a abordagem anteriormente apresentada.

Tabela 4.1: Dados coletados com o robô ARAMIS em movimento.

Primeira coleta de dados						
Leitora 10	Leitora 11	Leitora 12	Leitora 13	Leitora 14	X(m)	Y(m)
106.3993	104.9793	99.5801	96.9595	100.5101	1.05	0.35
115.3058	110.7402	101.3988	98.4681	105.5788	1.05	0.85
120.9447	112.3003	101.6366	105.5764	111.1650	1.05	1.55
121.5593	109.9569	101.2091	109.9505	115.6259	1.05	2.15
121.7325	111.8157	97.7910	112.7215	116.7800	1.05	2.75
Segunda coleta de dados						
Leitora 10	Leitora 11	Leitora 12	Leitora 13	Leitora 14	X(m)	Y(m)
97.7761	102.7746	99.9835	98.4394	101.6138	1.05	0.35
103.0725	104.4669	101.1916	97.0134	103.0491	1.05	0.95
106.2083	104.8513	101.5114	95.5993	102.8076	1.05	1.55
108.4887	104.6886	101.7038	95.0183	102.5383	1.05	2.15
109.255	101.3743	98.2861	99.1627	105.5461	1.05	2.75

Tabela 4.2: Dados coletados para um ponto em X e Y

Leitora 10	Leitora 11	Leitora 12	Leitora 13	Leitora 14 (%)
111	100	70	100	102
111	100	88	99	101
111	98	88	100	100
110	100	89	100	100
110	100	89	98	102

Como pode ser visto nos dados apresentados, uma repetibilidade das condições de treinamento da rede neural é praticamente impossível, logo a localização é prejudicada. Uma alternativa a ser analisada é colocar-se *tags* fixas no laboratório e com suas respectivas posições sendo conhecidas se houvesse variação em seus RSSI o sistema de localização atualizaria em tempo real seus pesos (da rede neural) para localizar com mais precisão o usuário.

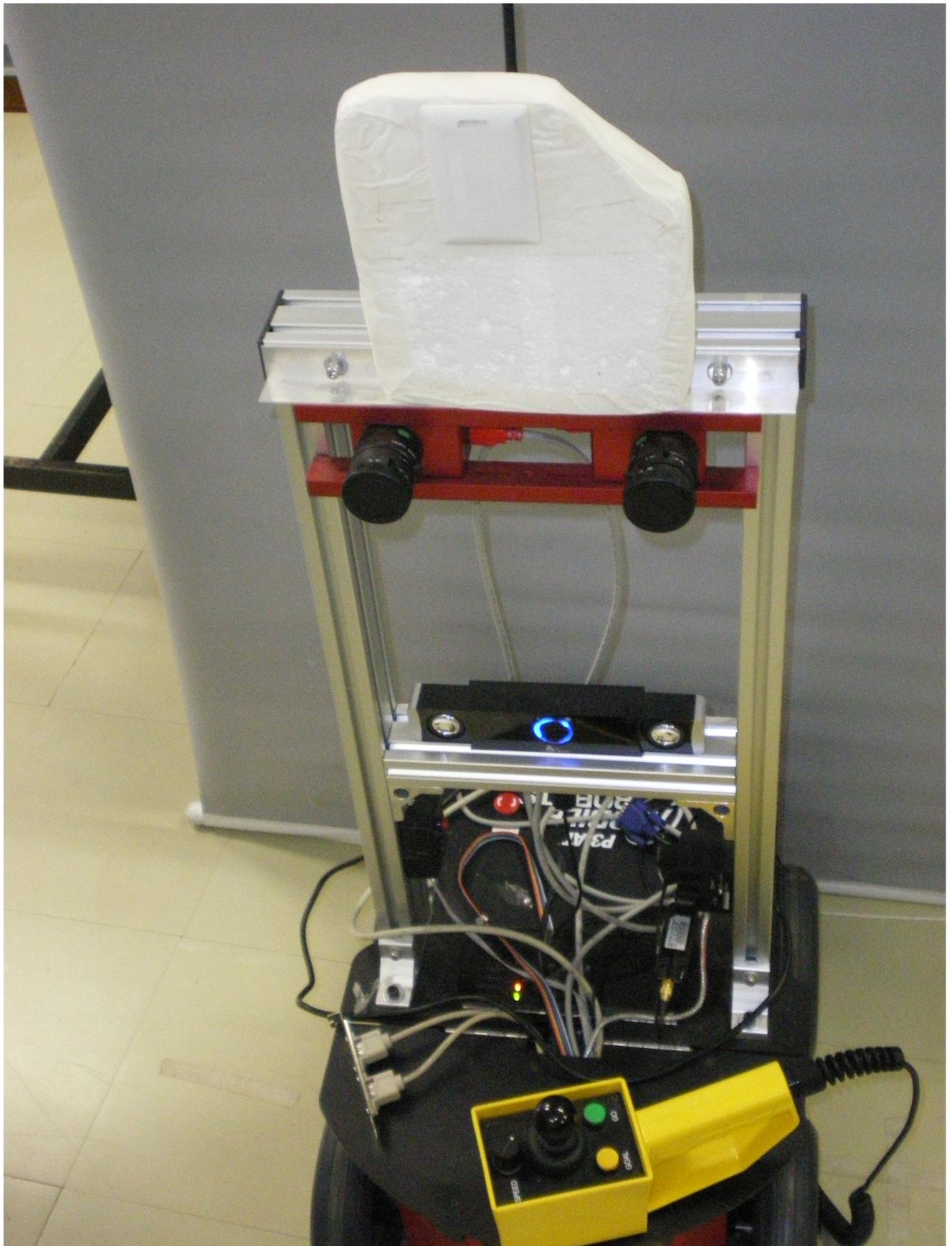


Figura 4.3: Robô móvel aramis.

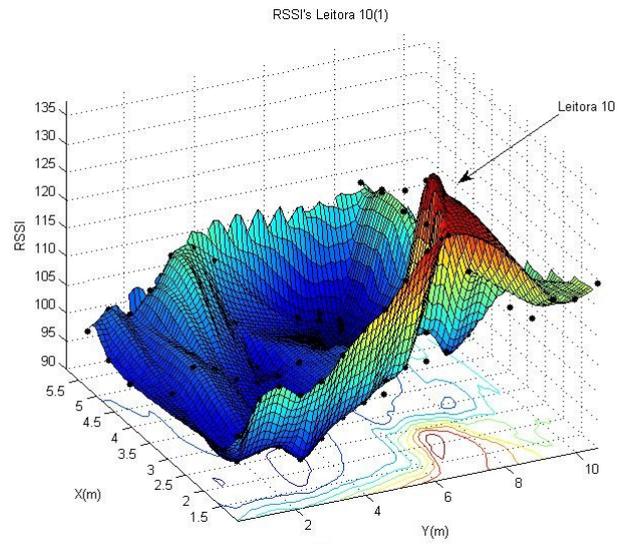


Figura 4.4: RSSIs coletados sem presença humana na Leitora 10.

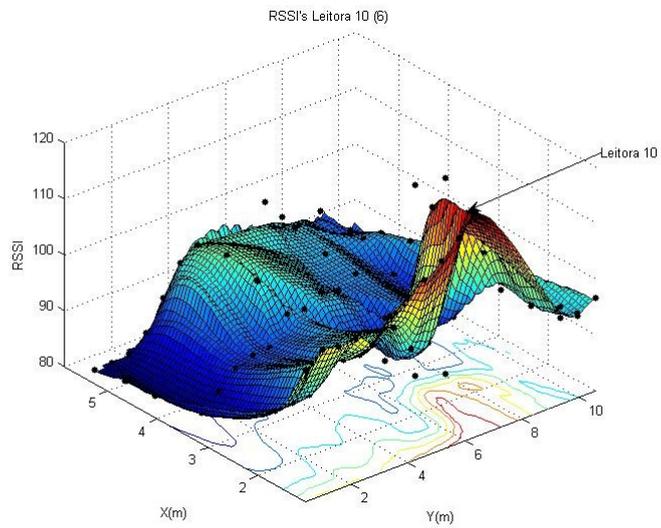


Figura 4.5: RSSIs coletados com presença humana na Leitora 10.

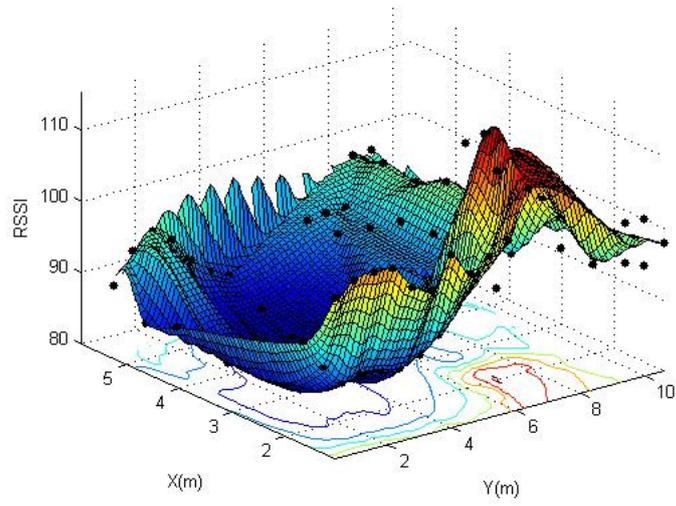


Figura 4.6: RSSIs coletados na Leitora 10 .

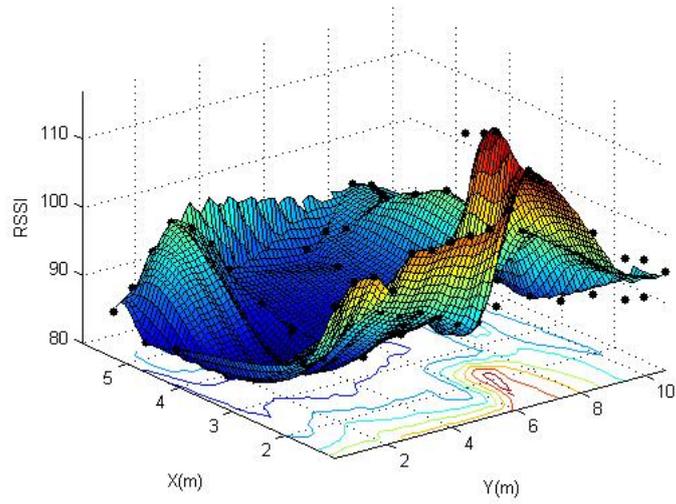


Figura 4.7: RSSIs coletados na Leitora 10.

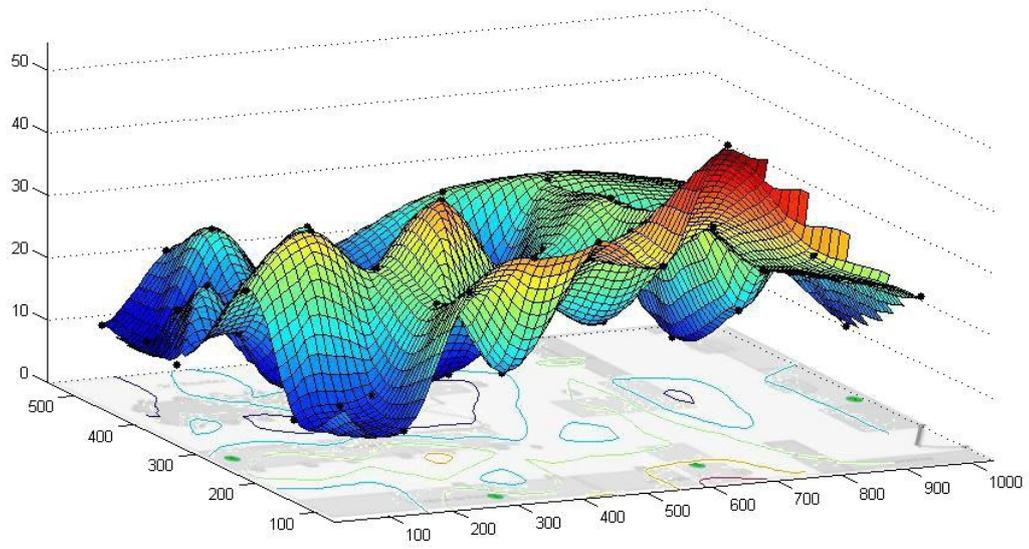


Figura 4.8: RSSIs coletados com a *tag* parada sobre a mesa [9].

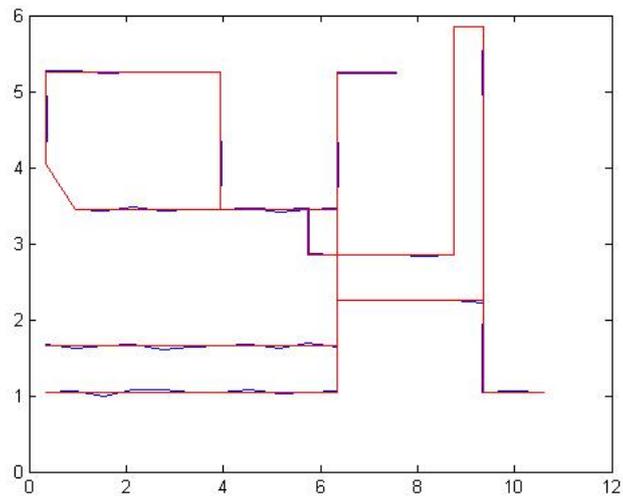


Figura 4.9: Saída X e Y da RNA esperada.

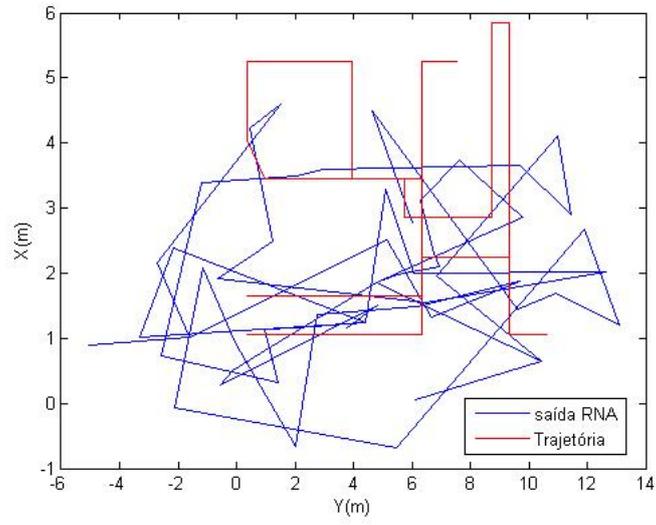


Figura 4.10: Saída X e Y da RNA treinada com os dados coletados discrepantes entre si.

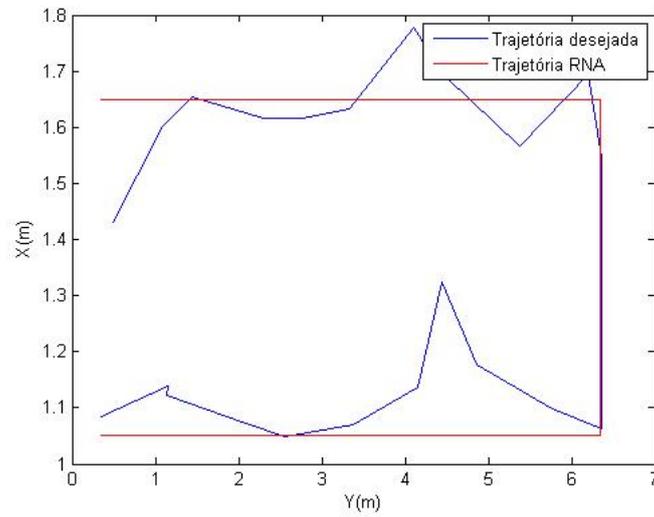


Figura 4.11: Resultado da RNA para um dos dados treinados.



Figura 4.12: Trajetória desenhada no chão do LARA.

Capítulo 5

Conclusões

Os objetivos propostos para o trabalho foram alcançados, a comunicação entre o supervisor ActionView e o módulo de aquisição de dados é estável, pois o servidor OPC do ActionView fica ativo durante todo o tempo em que se exigir um supervisionamento do ambiente, sendo assim qualquer cliente OPC pode ler e escrever (*read/write access*) a todo momento, tendo-se aplicações variadas de um mesmo dispositivo, no caso o RFID. Por vezes a comunicação TCP/IP com as leitoras RFID era perdida e a única maneira de restabelecê-la era desligando e ligando as leitoras. Porém, esse fato não chegou a limitar o progresso do trabalho, sendo somente um fator a ser relevado em outra implementação no futuro.

A etapa de acionamento dos aparelhos de ar condicionado não foi possível com os módulos de comunicação ZigBee sobre BACNet implementado em trabalhos anteriores pois não funcionavam e não havia documentação técnica dos mesmos. Optou-se então por uma comunicação via ModBus, com um módulo que já dispunha deste protocolo, sendo assim o acionamento foi feito com fios e funciona sem defeitos. Sendo factível em qualquer ambiente em que se deseje um acionamento deste tipo.

Ao utilizar-se RFID para localização, percebeu-se que os sinais de potência RSSI enviados não decaíam com o inverso da distância ao quadrado, os múltiplos caminhos que os dados enviados pela *tag* percorriam atrapalham no sistema de localização implementado.

Recomenda-se para trabalhos futuros, que se compre *tags* padronizadas, mais robustas, omnidirecionais e que enviem sinais com uma frequência tal que não sofra tanta interferência de qualquer obstáculo físico que se interponha em seu caminho, pois foi observado que os valores de RSSI variavam muito de *tag* para *tag*, logo fica impossibilitado de se fazer modelagem dos sinais para uma implementação de filtros de Kalman mais eficientes, tanto na entrada, quanto na saída X,Y da rede neural. Uma proposta interessante para um desenvolvimento futuro é colocar *tags* fixas no laboratório que possam atualizar em tempo real os pesos da rede neural, mas deve-se sempre ter em mente que se não forem adquiridas *tags* padronizadas, ou seja, que a uma mesma distância transmitam um mesmo RSSI, ou ao menos não variem tanto de *tag* para *tag*, com uma abordagem deste tipo a interferência de pessoas nos ambientes poderia ser compensada ajustando-se os pesos dos neurônios da RNA. Para obter-se resultados mais precisos, pode-se implementar um filtro de

Kalman na saída da RNA, que funcionaria como um filtro passa baixas e não deixaria a posição do usuário variar muito em intervalos de tempo pequenos, não foi possível implementar este filtro de Kalman por falta de tempo para programar um modelo preciso da movimentação de pessoas dentro do Laboratório.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Krecké, E.D., Morais, G., Santini, R. *Sistema de climatização natural, soluções com a tecnologia isomax para a segurança energética do Brasil*, 2009.
- [2] ANEEL, Agência Nacional de Energia Elétrica, *Atlas de energia elétrica do Brasil*, 1ª Edição, 2002.
- [3] SPIN Engenharia de automação, Manual ActionView de protocolos, *Revisão A*, 2008.
- [4] BAUCHSPIESS, A. *Introdução aos Sistemas Inteligentes Aplicações em Engenharia de Redes Neurais Artificiais, Lógica Fuzzy e Sistemas Neuro-Fuzzy*, 2008.
- [5] XAVIER, A. A. P., LAMBERTS, R., VOLPATO, C.G. *Relacionamento entre sensações térmicas e percentagem de pessoas insatisfeitas com o ambiente em 38 estudos de campo*, *ENTAC 2000 - VIII Encontro Nacional de Tecnologia do Ambiente Construído - Modernidade e Sustentabilidade*, Salvador/BA. Laboratório de eficiência energética em edificações. Universidade Federal de Santa Catarina, Florianópolis/SC, 2000.
- [6] GALLO, E. A., RIBEIRO, F. N. *Índice de conforto térmico ISO 7730 em Automação Predial*. Trabalho de Graduação em Engenharia de Controle e Automação, Faculdade de Tecnologia, Universidade de Brasília, 2007.
- [7] KRUTZ, R. L. *Securing SCADA SYSTEMS*. Wiley Publishing, 2006.
- [8] NOVAIS, M. C. C. e DUARTE J. U. J. *Instrumentação e controle de um sistema de ar condicionado híbrido utilizando BACNet sobre ZigBee*. Trabalho de Graduação em Engenharia de Controle e Automação, Faculdade de Tecnologia, Universidade de Brasília, 2010.
- [9] FONSECA, L. O. *Sistema de localização RFID de usuários visando racionalização de energia em ambientes inteligentes*. Trabalho de Graduação em Engenharia de Controle e Automação, Faculdade de Tecnologia, Universidade de Brasília, 2011.
- [10] BORGES, L. P. e DORES, R. C. *Automação predial sem fio utilizando BACnet/ZIGBEE com foco em economia de energia*. Trabalho de Graduação em Engenharia de Controle e Automação, Faculdade de Tecnologia, Universidade de Brasília, 2010.
- [11] SANTANA, P. H. R. Q. *Filtragem estocástica para sistemas híbridos e suas aplicações em robótica aérea*. Dissertação de Mestrado, Faculdade de Tecnologia, Universidade de Brasília, 2011.

- [12] EVANGELISTA, D. S. *Integração de redes de sensores ZigBee para automação predial utilizando módulos Meshbean*. Trabalho de Graduação em Engenharia Elétrica, Faculdade de Tecnologia, Universidade de Brasília, 2010.
- [13] COUTO, F. L. e FIGUEREDO, L. F. da C. *Medição móvel de conforto térmico para rede de Automação Predial Wireless*. Trabalho de Graduação em Engenharia de Controle e Automação, Faculdade de Tecnologia, Universidade de Brasília, 2008.
- [14] CARDOSO, M. W. A. *Sistema supervisorio em ActionView para implementação de conforto térmico em rede Wireless*. Trabalho de Graduação em Engenharia de Controle e Automação, Faculdade de Tecnologia, Universidade de Brasília, 2010.
- [15] MIRANDA, A. C. A. e FREITAS, H. L. P. de C. *Projeto e automação de um sistema de ar condicionado híbrido*. Trabalho de Graduação em Engenharia de Controle e Automação, Faculdade de Tecnologia, Universidade de Brasília, 2010.

ANEXOS

A.1 Redes Neurais Artificiais

Rumelhart, Hinton e Williams, 1986

Dinâmica: EP- Elemento Processador (neurônio).

$$s_j^{(k)} = w_j^{(k)} + \sum_i w_{ij}^k x_i^{(k-1)}.$$

$x + j^{(k)} = f(s_j^{(k)})$ com f continua diferencial.

Treinamento

$$\epsilon^2 = \sum_{j=1} (d_j - y_j)^2 \text{-erro quadrático,}$$

$W_j^{(k)} = (w_{0j}^{(k)}, w_{1j}^{(k)}, \dots, w_{mj}^{(k)})$ -vetor de pesos do EP $_j$,

$X_j^{(k-1)} = (1, x_{1j}^{(k-1)}, \dots, x_{nj}^{(k-1)})$ -vetor de entradas do EP $_j$.

Gradiente instantâneo:

$$\nabla_j^{(k)} = \frac{\partial \epsilon^2}{\partial w_j^{(k)}} = \left[\frac{\partial \epsilon^2}{\partial w_{0j}^{(k)}}, \frac{\partial \epsilon^2}{\partial w_{1j}^{(k)}}, \dots, \frac{\partial \epsilon^2}{\partial w_{mj}^{(k)}} \right],$$

$$\nabla_j^{(k)} = \frac{\partial \epsilon^2}{\partial w_j^{(k)}} = \frac{\partial \epsilon^2}{\partial s_j^{(k)}} \frac{\partial s_j^{(k)}}{\partial w_j^{(k)}},$$

como $s_j^{(k)} = w_j^{(k)} x_j^{(k-1)}$.

$$\frac{\partial s_j^{(k)}}{\partial w_j^{(k)}} = x_j^{(k-1)},$$

e portanto

$$\nabla_j^{(k)} = \frac{\partial \epsilon^2}{\partial w_j^{(k)}} = \frac{\partial s_j^{(k)}}{\partial w_j^{(k)}} x_j^{(k-1)}.$$

O **erro derivativo quadrático** é definido por $\delta_j^{(k)} = \frac{-1}{2} \frac{\delta \epsilon^2}{\delta s_j^{(k)}}$.

$$\nabla_j^{(k)} = -2\delta_j^{(k)} x_j^{(k-1)}.$$

Para a camada de saída, o erro derivativo quadrático é:

$$\delta_j^{(k)} = -\frac{1}{2} \frac{\partial \sum_{i=1}^{N_k} (d_i - y_i)^2}{\partial s_j^{(k)}} = -\frac{1}{2} \frac{\partial \sum_{i=1}^{N_k} (d_i - f(s_i^{(k)}))^2}{\partial s_j^{(k)}},$$

e como as derivadas parciais para $i \neq j$ se anulam

$$\partial_j^{(k)} = -\frac{1}{2} \frac{\partial (d_j - f(s_j^{(k)}))^2}{\partial s_j^{(k)}} = -(d_j - f(s_j^{(k)})) \frac{\partial (d_j - f(s_j^{(k)}))}{\partial s_j^{(k)}} = (d_j - x_j^{(k)}) f'(s_j^{(k)}).$$

Erro na saída associado ao EP_j da última camada:

$$\epsilon_j^{(k)} = d_j - x_j^{(k)} = d_j - y_j,$$

$$\text{implicando em } \partial_j^{(k)} = \epsilon_j^{(k)} f'(s_j^{(k)}).$$

Desenvolvimento para uma camada oculta (k): O erro quadrático da camada k é determinado pelas saídas lineares da camada $k+1$.

$$\delta_j^{(k)} = -\frac{1}{2} \frac{\delta \epsilon^2}{\delta s_j^{(k)}}.$$

desenvolvendo a última equação tem-se:

$$\delta_{i^{(k)}} = \sum_{i=1}^{N_{k+1}} (\delta_i^{(k+1)} w_{ji}^{(k+1)} f'(s_j^{(k)})).$$

A atualização dos pesos é feita por:

$$W_j^{(k)}(n+1) = W_j^{(k)}(n) + 2\mu \delta_j^{(k)}(n) X_j^{(k-1)}(n). \quad (\text{A.1})$$

A.2 Algoritmo *Backpropagation*

1. $w_{ij}^{(k)} \leftarrow \text{random}$, inicializar a rede,
2. $p/(x, d)$, par de treinamento, obter y . Propagação *feedforward*

$$\epsilon^2 = \sum_{j=1}^m (d_j - y_j)^2.$$

3. $k \leftarrow$ última camda,
4. para todo elemento j da camada k faça:

Calcule $\epsilon_j^{(k)}$ empregando $\epsilon_j^{(k)} = (d_j - y_j)$ se k , for a última camada,

$$\epsilon_j^{(k)} = \sum_{i=1}^{N_{k+1}} \delta_i^{(k+1)} w_{ji}^{(k+1)}.$$

se for uma camada oculta.

Calcule $\delta_j^{(k)} = \epsilon_j^{(k)} \cdot f'(s_j^{(k)})$,

5. $k \leftarrow k - 1$ se $k > 0$ vá para o passo 4, senão prossiga.
6. $W_j^{(k)}(n+1) = W_j^{(k)}(n) + 2\mu \delta_j^{(k)}(n) X_j^{(n-1)}(n)$.
7. para o próximo par de treinamento vá para o passo 2.

O algoritmo *Error Backpropagation* levou a uma grande aceitação das RNAs por parte da comunidade científica, uma vez que redes multicamadas podem ser treinadas a partir dos dados que representam amostras significativas do processo em questão [4].

A.3 Considerações práticas sobre o algoritmo *Backpropagation*.

É em geral muito lento e suscetível a “patologias” de treinamento. A parada da rede ocorre em regiões de gradiente próximos de zero (platôs). Dependendo das condições iniciais o treinamento fica preso em mínimos locais da superfície de erro.

O critério de parada do treinamento considera em geral um limite máximo do número de épocas de treinamento. Além disso o treinamento pode terminar quando a soma do erro quadrático ou a média deste atinge o seu objetivo. Para acelerar a convergência utilizam-se variantes do *Backpropagation*. A função *trainbpm* (com momento, mC) considera não apenas o gradiente local, mas também tendências recentes da superfície de erro. Atua como um filtro passa-baixas, ignorando características menores (pontos de mínimo local) da superfície de erro. O treinamento “desliza ” sobre mínimos locais não muito pronunciados.

B. COMO CRIAR VARIÁVEIS

Primeiramente definiu-se uma regional que no projeto foi nomeada SERVIÇOS GERAIS 11, dentro dessa regional pode-se ter vários sistemas, para criar um sistema basta clicar com o botão direito no diretório da regional mostrado na tela árvore de projeto, conforme a Figura B.1, posteriormente sugerirá uma configuração para o sistema (Figura B.2).



Figura B.1: Configuração da regional no ActionView.

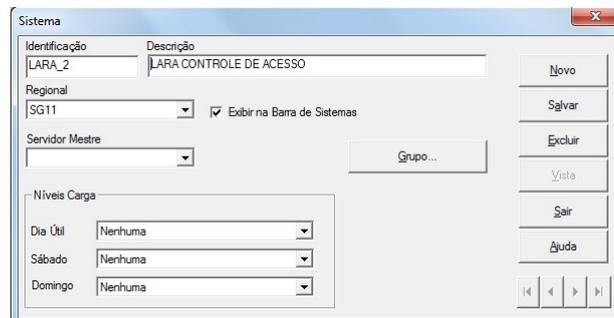


Figura B.2: Configuração dos parâmetros do Sistema.

Depois de criado o sistema, precisa-se criar um grupo, clicando com o botão direito no sistema criado, clique em adicionar grupo (Figura B.3), gerando uma tela auxiliar para a configuração dos parâmetros do grupo conforme a Figura B.4

Quando um grupo é criado, ele já possui dentro de si variáveis analógicas, digitais, digitais múltiplas, funções e condições (Figura B.5). Dependendo da aplicabilidade do usuário, faz-se a devida escolha para cada variável que se necessita. No projeto foram utilizadas diversas variáveis, para testes, validação de comunicações, verificação de entradas e saídas do sistema, conforme a Figura B.6. As variáveis do tipo SD, são as saídas do sistema, elas são configuradas para que acionem o ar-condicionado, como no projeto foi utilizado o protótipo Es 1050 com oito entradas e oito saídas, só foram criadas oito saídas, configuradas conforme mostrado na Figura B.7, sendo do tipo Variáveis digitais múltiplas, as que vão para o ES 1050, e as variáveis digitais, são utilizadas



Figura B.3: Adicionando um grupo ao Sistema .

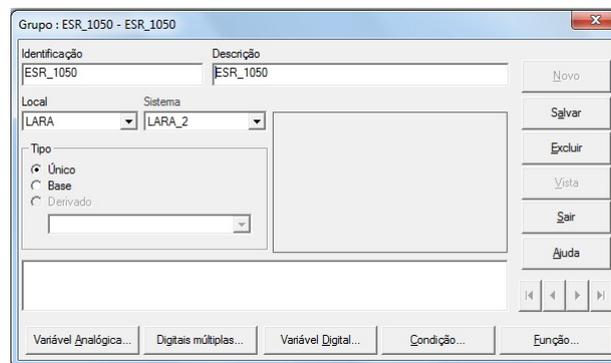


Figura B.4: Configuração dos parâmetros do grupo.

internamente, para se mudar o estado do ar condicionado na tela do ActionView de On para OFF ou OFF para ON.

As variáveis analógicas no grupo VARIÁVEIS_ACESSO são de uso interno para o script e para amostragem na tela do supervisor.

B.1 Script ActionView

```

Sub OnTimer()
' converte o que ven do campo
DIM VAR1_X
DIM VAR1_Y
DIM Val_Ligar DIM VAR2_X DIM VAR2_Y

```

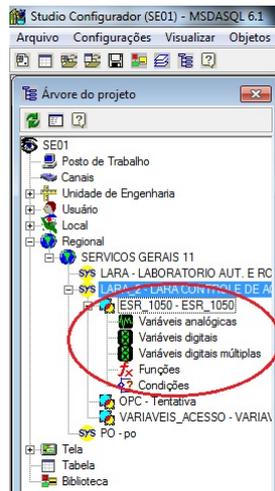


Figura B.5: Variáveis iniciais do grupo criado.

```

'VAR1_X = Var.Value("VARIAVEIS_ACESSO", "POS1_X") + 10
'VAR1_Y = Var.Value("VARIAVEIS_ACESSO", "POS1_Y") + 10
*****COM O OPC AGORA*****
Var.Value("VARIAVEIS_ACESSO", "POS1_Y") = Var.Value("OPC","POS_Y")
Var.Value("VARIAVEIS_ACESSO", "POS1_X") = Var.Value("OPC","POS_X")
'Cálculo da posicao na tela, foi feito através do mapeamento dos pontos pixels no modo
' runtime com 100% de exibicao, achando assim os valores corresponde
'a metragem original do LARA
Controls.Item("LARA|Digital0").Top = (-Var.Value("VARIAVEIS_ACESSO", "POS1_Y")
+ 11)*1050.090909
Controls.Item("LARA|Digital0").Left = (-Var.Value("VARIAVEIS_ACESSO", "POS1_X")
+ 18)*1000.5
Val_Ligar=6 'variável que define quanto tempo a pessoa necessita estar no ambiente para
acionar o ar-condicionado.
' comanda ventilador
' variavel que habilita comando do ventilador
if Var.Value("VARIAVEIS_ACESSO", "POS1_X") < 5 and Var.Value("VARIAVEIS_ACESSO",
"POS1_X") > 2
and Var.Value("VARIAVEIS_ACESSO", "POS1_Y") < 5 then
Var.Value("VARIAVEIS_ACESSO", "AMB2")=0
Var.Value("VARIAVEIS_ACESSO", "AMB3")=0

```

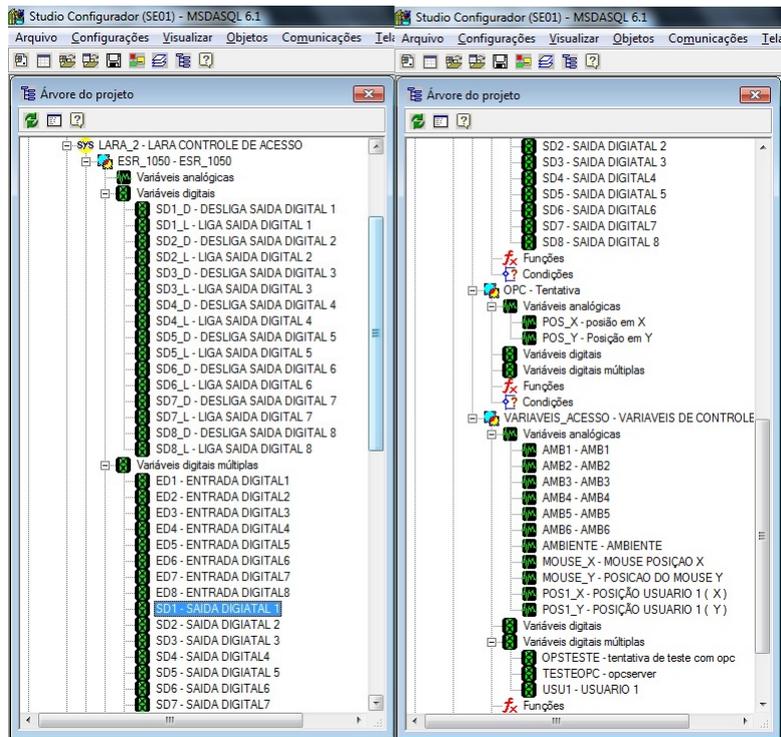


Figura B.6: Variáveis utilizadas no projeto.

```

Var.Value("VARIAVEIS_ACESSO", "AMB4")=0
Var.Value("VARIAVEIS_ACESSO", "AMB5")=0
Var.Value("VARIAVEIS_ACESSO", "AMB6")=0
Var.Value("VARIAVEIS_ACESSO", "AMB1")= Var.Value("VARIAVEIS_ACESSO", "AMB1")+1
'CONTADOR PARA LGAR O ARCONDICIONADO 1 DEPOIS DE X SEGUNDOS, NO
CASO X=6
Var.Value("VARIAVEIS_ACESSO", "AMBIENTE")= Var.Value("VARIAVEIS_ACESSO",
"AMB1")
if Var.Value("VARIAVEIS_ACESSO", "AMB1")=Val_Ligar then
Call Command("ESR_1050","SD1_L",6,,Var.CommandParameter("ESR_1050","SD1_L"),false)
end if
else
Call Command("ESR_1050","SD1_D",6,,Var.CommandParameter("ESR_1050","SD1_D"),false)
end if
if Var.Value("VARIAVEIS_ACESSO", "POS1_X")< 2 and Var.Value("VARIAVEIS_ACESSO",
"POS1_Y")< 5 then
Call Command ("ESR_1050","SD2_L",6,"ESR_1050","SD2_L",Var.CommandParameter("ESR_1050",S

```

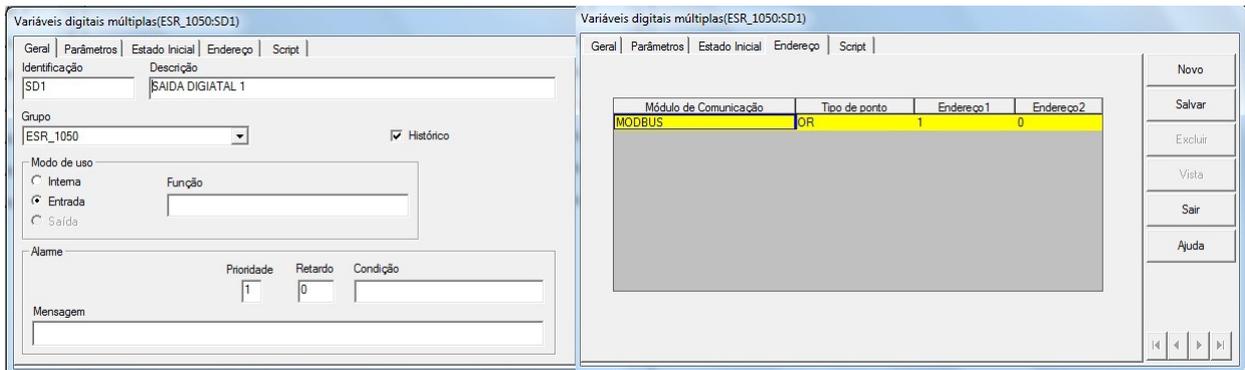


Figura B.7: Configuração dos parâmetros da variável digital múltipla.

```

Var.Value("VARIABLES_ACESSO", "AMB1")=0
Var.Value("VARIABLES_ACESSO", "AMB3")=0
Var.Value("VARIABLES_ACESSO", "AMB4")=0
Var.Value("VARIABLES_ACESSO", "AMB5")=0
Var.Value("VARIABLES_ACESSO", "AMB6")=0
Var.Value("VARIABLES_ACESSO", "AMB2")= Var.Value("VARIABLES_ACESSO", "AMB2")+1
'CONTADOR PARA LGAR O ARCONDICIONADO 2 DEPOIS DE X SEGUNDOS, NO
CASO X=6
Var.Value("VARIABLES_ACESSO", "AMBIENTE")= Var.Value("VARIABLES_ACESSO",
"AMB2")
if Var.Value("VARIABLES_ACESSO", "AMB2")=Val_Ligar then
Call Command ("ESR_1050","SD2_L",6,,,Var.CommandParameter("ESR_1050","SD2_L"),FALSE)
end if
else
Call Command("ESR_1050","SD2_D",6,,,Var.CommandParameter("ESR_1050","SD2_D"),false)
end if
if Var.Value("VARIABLES_ACESSO", "POS1_X")< 5 and Var.Value("VARIABLES_ACESSO",
"POS1_Y")> 5
and Var.Value("VARIABLES_ACESSO", "POS1_Y") < 11 then
Var.Value("VARIABLES_ACESSO", "AMB1")=0
Var.Value("VARIABLES_ACESSO", "AMB2")=0
Var.Value("VARIABLES_ACESSO", "AMB4")=0
Var.Value("VARIABLES_ACESSO", "AMB5")=0

```

```

Var.Value("VARIABLES_ACESSO", "AMB6")=0
Var.Value("VARIABLES_ACESSO", "AMB3")=Var.Value("VARIABLES_ACESSO", "AMB3")+1
'CONTADOR PARA LGAR O ARCONDICIONADO 3 DEPOIS DE X SEGUNDOS, NO
CASO X=6
Var.Value("VARIABLES_ACESSO", "AMBIENTE")= Var.Value("VARIABLES_ACESSO",
"AMB3")
if Var.Value("VARIABLES_ACESSO", "AMB3")=Val_Ligar then
Call Command("ESR_1050","SD3_L",6,,Var.CommandParameter("ESR_1050","SD3_L"),false)
end if
else
if Var.CommandParameter("ESR_1050","SD3_L") then
Call Command("ESR_1050","SD3_D",6,,Var.CommandParameter("ESR_1050","SD3_D"),false)
end if
end if
if Var.Value("VARIABLES_ACESSO", "POS1_X")> 5 and Var.Value("VARIABLES_ACESSO",
"POS1_Y")> 5
and Var.Value("VARIABLES_ACESSO", "POS1_Y") < 11 and Var.Value("VARIABLES_ACESSO",
"POS1_Y") < 9.6 then
'Call Command ("ESR_1050","SD4_L",6,"ESR_1050","SD1_D",Var.CommandParameter,FALSE)
Var.Value("VARIABLES_ACESSO", "AMB1")=0
Var.Value("VARIABLES_ACESSO", "AMB2")=0
Var.Value("VARIABLES_ACESSO", "AMB3")=0
Var.Value("VARIABLES_ACESSO", "AMB5")=0
Var.Value("VARIABLES_ACESSO", "AMB6")=0
Var.Value("VARIABLES_ACESSO", "AMB4")=Var.Value("VARIABLES_ACESSO", "AMB4")+1
Var.Value("VARIABLES_ACESSO", "AMBIENTE")= Var.Value("VARIABLES_ACESSO",
"AMB4")
if Var.Value("VARIABLES_ACESSO", "AMB4")=Val_Ligar then
Call Command("ESR_1050","SD4_L",6,,Var.CommandParameter("ESR_1050","SD4_L"),false)
end if
else
if Var.CommandParameter("ESR_1050","SD4_L") then

```

```

Call Command("ESR_1050","SD4_D",6,,Var.CommandParameter("ESR_1050","SD4_D"),false)
end if
end if
if (Var.Value("VARIABLES_ACESSO", "POS1_X") > 9.5 and Var.Value("VARIABLES_ACESSO",
"POS1_Y") < 11
and Var.Value("VARIABLES_ACESSO", "POS1_X") < 12.5) or (Var.Value("VARIABLES_ACESSO",
"POS1_Y") < 5
and Var.Value("VARIABLES_ACESSO", "POS1_X") < 12.5 and Var.Value("VARIABLES_ACESSO",
"POS1_Y")) >5 then
Var.Value("VARIABLES_ACESSO", "AMB1")=0
Var.Value("VARIABLES_ACESSO", "AMB2")=0
Var.Value("VARIABLES_ACESSO", "AMB3")=0
Var.Value("VARIABLES_ACESSO", "AMB4")=0
Var.Value("VARIABLES_ACESSO", "AMB6")=0
Var.Value("VARIABLES_ACESSO", "AMB5")=Var.Value("VARIABLES_ACESSO", "AMB5")+1
'CONTADOR PARA LGAR O ARCONDICIONADO 3 DEPOIS DE X SEGUNDOS, NO
CASO X=6
Var.Value("VARIABLES_ACESSO", "AMBIENTE")= Var.Value("VARIABLES_ACESSO",
"AMB5")
if Var.Value("VARIABLES_ACESSO", "AMB5")=Val_Ligar then
Call Command("ESR_1050","SD5_L",6,,Var.CommandParameter("ESR_1050","SD5_L"),false)
end if
else
if Var.CommandParameter("ESR_1050","SD5_L") then
Call Command("ESR_1050","SD5_D",6,,Var.CommandParameter("ESR_1050","SD5_D"),false)
end if
end if
if (Var.Value("VARIABLES_ACESSO", "POS1_X") < 18 and Var.Value("VARIABLES_ACESSO",
"POS1_Y") < 11
and Var.Value("VARIABLES_ACESSO", "POS1_X") > 12.5 and Var.Value("VARIABLES_ACESSO",
"POS1_Y") >5)then
Var.Value("VARIABLES_ACESSO", "AMB1")=0
Var.Value("VARIABLES_ACESSO", "AMB2")=0

```

```

Var.Value("VARIABLES_ACESSO", "AMB3")=0
Var.Value("VARIABLES_ACESSO", "AMB4")=0
Var.Value("VARIABLES_ACESSO", "AMB5")=0
Var.Value("VARIABLES_ACESSO", "AMB6")=Var.Value("VARIABLES_ACESSO", "AMB6")+1
'CONTADOR PARA LGAR O ARCONDICIONADO 6 DEPOIS DE X SEGUNDOS, NO
CASO X=6
Var.Value("VARIABLES_ACESSO", "AMBIENTE")= Var.Value("VARIABLES_ACESSO",
"AMB6")
if Var.Value("VARIABLES_ACESSO", "AMB6")=Val_Ligar then
Call Command("ESR_1050","SD6_L",6,,,Var.CommandParameter("ESR_1050","SD6_L"),false)
end if

else
if Var.CommandParameter("ESR_1050","SD6_L") then
Call Command("ESR_1050","SD6_D",6,,,Var.CommandParameter("ESR_1050","SD6_D"),false)
end if
end if
End sub

```

C. PUTTY

Os dados das leitoras são transmitidos via TCP/IP, pela Ethernet, porém o acesso a essas informações fica restrito a rede interna aonde foram instaladas as mesmas, o que dificulta o trabalho feito fora da universidade. Tendo em mente a questão de segurança da informação da rede interna do LARA, instalou-se o programa PUTTY (*programa de acesso remoto a redes internas*), com um login e uma senha para tal acesso, que permitem que o 'localhost' da máquina, aonde quer que esteja, seja o do LARA.

Para uma melhor compreensão de como foi feito esse tipo de acesso, quais as configurações que se deva possuir, algumas orientações são citadas abaixo:

- Primeiramente deve-se realizar o download do programa. Uma página na internet que possui o programa para download é a <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.htm>.
- Fazer um login na rede interna do local onde as leitoras se encontram, o que só pode ser feito pelo administrador da rede.
- Configurar o programa Putty para acessar a rede, conforme as Figuras C.1, C.2 e C.3 nos mostram.

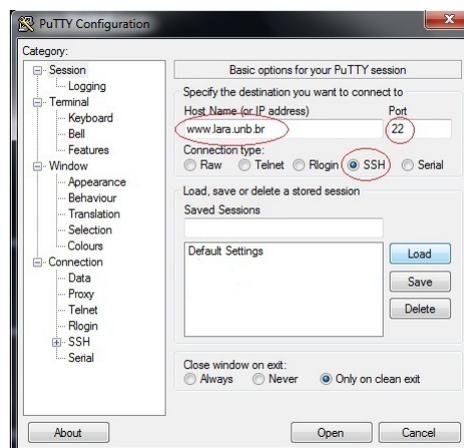


Figura C.1: Configuração dos parâmetros iniciais do Putty.

No caso da rede do LARA, as configurações devem ser especificamente essas, devido ao fato do HOST ser `www.lara.unb.br` através da porta 22 (Figura C.1), e as Leitoras das TAGS de RFID se encontrarem nos endereços internos especificados. Como no trabalho foram usadas 5 leitoras, seus respectivos endereços de rede são: 192.168.0.10, 192.168.0.11, 192.168.0.12, 192.168.0.13, 192.168.0.14 se encontrando na porta interna 10001. Para se ter acesso a essas leitoras foi especificada uma 'Porta virtual' para elas, a partir da 5000 até a 5004, o que não torna esses valores uma regra, apenas uma escolha (Figura C.2). Após a conexão ser

estabelecida o usuário irá visualizar um terminal, que é o terminal de acesso da rede interna, no exemplo feito é a do LARA, conforme a Figura C.3 nos mostra.

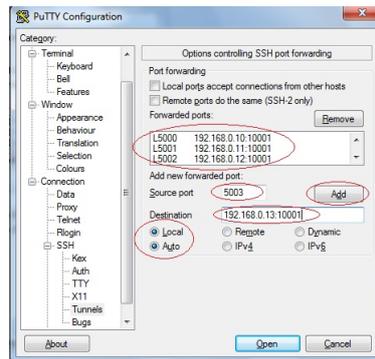


Figura C.2: Configuração dos parâmetros intermediários do Putty.

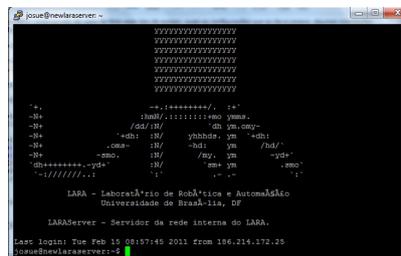


Figura C.3: Tela final após conexão estabelecida.

Para se fazer um teste verificando se realmente os dados das leitoras estão sendo enviados pela conexão feita no PUTTY, fez-se um programa no Matlab, conforme as linha de comando descritas abaixo: `Leitora_10=tcPIP('127.0.0.1',5000);`

`fopen(Leitora_10);`

`Dado_Leitora_10=fread(Leitora_10,39);`

`fclose(Leitora_10);`

Necessitando-se apenas modificar o segundo parâmetro da função `tcPIP` de 5000 para 5001, 5002, 5003 ou 5004, conforme especificado nas configurações do PUTTY.

D. CÓDIGO MATLAB

D.1 Código leitura de dados

```
%leitora que está no IP 192.168.0.10
Leitora_10 = tcpip('192.168.0.10',10001);
fopen(Leitora_10);
%leitora que está no IP 192.168.0.11
Leitora_11 = tcpip('192.168.0.11',10001);
fopen(Leitora_11);
%leitora que está no IP 192.168.0.12
Leitora_12 = tcpip('192.168.0.12',10001);
fopen(Leitora_12);
%leitora que está no IP 192.168.0.13
Leitora_13 = tcpip('192.168.0.13',10001);
fopen(Leitora_13);
%leitora que está no IP 192.168.0.14
Leitora_14 = tcpip('192.168.0.14',10001);
fopen(Leitora_14);
%xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx FK xxxxxxxxxxxxxxxx
n=5; %number of state
q=0.5; %std of process
r=2; %std of measurement
Q = q^2 * eye(n); % covariance of process
R = r^2 * eye(n); % covariance of measurement
f=@(x)[x(1);x(2);x(3);x(4);x(5)]; % nonlinear state equations
h=@(x)[x(1);x(2);x(3);x(4);x(5)]; % measurement equation
s=[89;99;97;96;95]; % initial state
x=s+q*randn(5,1); %initial state with noise
P = eye(n); % initial state covraiance
```

```

RSSI_Leitora_10_Tag_180=70;
RSSI_Leitora_11_Tag_180=70;
RSSI_Leitora_12_Tag_180=70;
RSSI_Leitora_13_Tag_180=70;
RSSI_Leitora_14_Tag_180=70;
fid3 = fopen('Dados_Leitora_Robot_real9.txt','a');
for k=1:91
echo on;
%press Enter to continue
pause;
echo off;
for i=1:2
%abaixo os if's estão separando os dados importantes em suas %respectivas variáveis de tags e
leituras no caso precisaremos de: %RSSI
Dados_leitora_10 = fread (Leitora_10,39);
Dados_leitora_11 = fread (Leitora_11,39);
Dados_leitora_12 = fread (Leitora_12,39);
Dados_leitora_13 = fread (Leitora_13,39);
Dados_leitora_14 = fread (Leitora_14,39);
if numel(Dados_leitora_10) ==0;
if Dados_leitora_10(26)== 180
RSSI_Leitora_10_Tag_180=Dados_leitora_10(29);
Age10 = Dados_leitora_10(19);
end
end
if numel(Dados_leitora_11) ==0
if Dados_leitora_11(26)== 180
RSSI_Leitora_11_Tag_180=Dados_leitora_11(29);
Age11 = Dados_leitora_11(19);
end
end
end

```

```

if numel(Dados_leitora_12) == 0
if Dados_leitora_12(26) == 180
RSSI_Leitora_12_Tag_180 = Dados_leitora_12(29);
Age12 = Dados_leitora_12(19);
end
end
if numel(Dados_leitora_13) == 0
if Dados_leitora_13(26) == 180
RSSI_Leitora_13_Tag_180 = Dados_leitora_13(29);
Age13 = Dados_leitora_13(19);
end
end
if numel(Dados_leitora_14) == 0
if Dados_leitora_14(26) == 180
RSSI_Leitora_14_Tag_180 = Dados_leitora_14(29);
Age14 = Dados_leitora_14(19);
end
end
Tag180 = [(RSSI_Leitora_14_Tag_180); (RSSI_Leitora_13_Tag_180); (RSSI_Leitora_12_Tag_180); (RSSI_Leitora_11_Tag_180)];
z = [(Age14); (Age13); (Age12); (Age11)];
c = clock;
z
[x, P] = ekf(f, x, P, h, z, Q, R);
x
pause(.1);
end
z;
x;
k
A(k)
B(k)

```

```

fprintf(fid3,\n%f \t%f \t%f \t%f\t%f \t %d:%d:%f \t %d \t %d \t %d \t %d \t %d,
x(1),x(2),x(3),x(4),x(5),c(4),c(5),c(6),A(k),B(k),AMB1(k),AMB2(k),AMB3(k));
%especificações da mensgaem enviada pelas tags
end
fclose(fid3)
%Fechamento da comunicação e delete das vairáveis utilizadas na comunicação %com as leitoras
fclose (Leitora_10);
delete(Leitora_10);
fclose (Leitora_11);
delete(Leitora_11);
fclose (Leitora_12);
delete(Leitora_12);
fclose (Leitora_13);
delete(Leitora_13);
fclose(Leitora_14);
delete(Leitora_14);

```

E. TABELA DOS PONTOS DE TREINAMENTO DA RNA

Tabela E.1: Tabela dos pontos X,Y de treinamento da RNA

X(m)	Y(m)	Ambiente 1	Ambiente 2	Ambiente 3
1.05	3.35	0	1	0
1.05	3.95	0	1	0
1.05	4.55	0	1	0
1.05	5.15	0	1	0
1.05	5.75	0	0	1
1.05	6.35	0	0	1
1.65	6.35	0	0	1
2.25	6.35	0	0	1
2.25	6.95	0	0	1
2.25	7.55	0	0	1
2.25	8.15	0	0	1
2.25	8.75	0	0	1
2.25	9.35	0	0	1
1.65	9.35	0	0	1
1.05	9.35	0	0	1
1.05	9.95	0	0	1
1.05	10.55	0	0	1
1.05	9.95	0	0	1
1.05	9.35	0	0	1
1.65	9.35	0	0	1
2.25	9.35	0	0	1
2.85	9.35	0	0	1
3.45	9.35	0	0	1
4.05	9.35	0	0	1
1.05	0.35	0	1	0
1.05	0.95	0	1	0
1.05	1.55	0	1	0
1.05	2.15	0	1	0
1.05	2.75	0	1	0
4.65	9.35	0	0	1
5.25	9.35	0	0	1

Tabela E.2: Tabela dos pontos X,Y de treinamento da RNA, continuação

X(m)	Y(m)	Ambiente 1	Ambiente 2	Ambiente 3
5.85	9.35	0	0	1
5.85	8.75	0	0	1
5.25	8.75	0	0	1
4.65	8.75	0	0	1
4.05	8.75	0	0	1
3.45	8.75	0	0	1
2.85	8.75	0	0	1
2.85	8.15	0	0	1
2.85	7.55	0	0	1
2.85	6.95	0	0	1
2.85	6.35	0	0	1
2.85	5.75	0	0	1
3.45	5.75	0	0	1
3.45	5.15	0	0	1
3.45	4.55	1	0	0
3.45	3.95	1	0	0
3.45	3.35	1	0	0
3.45	2.75	1	0	0
3.45	2.15	1	0	0
3.45	1.55	1	0	0
3.45	0.95	1	0	0
4.05	0.35	1	0	0
4.65	0.35	1	0	0
5.25	0.35	1	0	0
5.25	0.95	1	0	0
5.25	1.55	1	0	0
5.25	2.15	1	0	0
5.25	2.75	1	0	0
5.25	3.35	1	0	0
5.25	3.95	1	0	0

Tabela E.3: Tabela dos pontos X,Y de treinamento da RNA, continuação 2

X(m)	Y(m)	Ambiente 1	Ambiente 2	Ambiente 3
4.65	3.95	1	0	0
4.05	3.95	1	0	0
3.45	3.95	1	0	0
3.45	4.55	1	0	0
3.45	5.15	0	0	1
3.45	5.75	0	0	1
3.45	6.35	0	0	1
4.05	6.35	0	0	1
4.65	6.350	0	0	1
5.25	6.35	0	0	1
5.25	6.95	0	0	1
5.25	7.55	0	0	1
5.25	6.95	0	0	1
5.25	6.35	0	0	1
4.65	6.35	0	0	1
4.05	6.35	0	0	1
3.45	6.35	0	0	1
2.85	6.35	0	0	1
2.25	6.35	0	0	1
1.65	6.35	0	0	1
1.65	5.75	0	0	1
1.65	5.15	0	1	0
1.65	4.554	0	1	0
1.65	3.95	0	1	0
1.65	3.35	0	1	0
1.65	2.75	0	1	0
1.65	2.15	0	1	0
1.65	1.55	0	1	0
1.65	0.95	0	1	0
1.65	0.35	0	1	0

F. TABELAS RFID

Tabela F.1: Lista de comandos fornecido pela Wavetrend

Value	Function	Expect Response
0x01	Enable Auto Polling	Continuous Data Stream
0x02	Disable Auto Polling	Reply Packet
0x03	Ping Reader	Reply Packet + Error Number
0x04	Set Network ID	Reply Packet
0x05	Set Reader ID	Reply Packet
0x06	Get Tag Packet	Tag Packet
0x07	Set RSSI Value	Reply Packet + RSSI
0x08	Get RSSI Value	Reply Packet
0x09	Set Site Code	Reply Packet
0x0A	Get Site Code	Reply Packet + Site Code
0x0B	Set Receiver Gain	Reply Packet
0x0C	Get Receiver Gain	Reply Packet + Gain
0x0D	Set Alarm Filter	Reply Packet
0x0E	Get Alarm Filter	Reply Packet + Status
0x0F	Get Number of Invalid Tags	Reply Packet + Counter
0x10	Get Supply Voltage	Reply Packet + Voltage
0x11	Start RF white noise calculation	Reply Packet
0x12	Get RF white noise result	Reply Packet + Result
0x13	Clear buffers	Reply Packet
0x14	Set Relay Status	Reply Packet
0x15	Get Relay Status	Reply Packet + Result
0x16	Get Input Status	Reply Packet + Result
0x17	Re-Initialise (Factory defaults)	Reply Packet
0x18	Set Heart Beat Message	Reply Packet
0x19	Get Heart Beat Message	Reply Packet + Result
0xFF	Get Version Information	Reply Packet + Version Data

Tabela F.2: Representação dos dados respondidos

Byte	Name	Data	Example	Description
1	Header	0x55	85	Header of 485 packet
2	No. Data Bytes	0x20	32	Number of bytes in Data part (Byte 6 to 37).
3	Network ID	0x01	01	Network identifier
4	Receiver ID	0x01	80	Actual value from reader that is stored.
5	Node ID	0x01	01	Position no of the reader on the reader network
6	Command ID	0x06	06	Command identifier linked to a response.
7	Header	Š!Š	33	Header. Fixed value
8	Header	Š*Š	42	Header. Fixed value
9	Header	Š*	42	Header. Fixed value
10	Interval		49	Time Interval between transmissions
		48	0x30	1 Transmission every 30 seconds
		49	0x31	1 Transmission every 1.5 seconds
		50	0x32	1 Transmission every 0.8 seconds
		51	0x33	1 Transmission every 0.4 seconds
		32	0x20	1 Transmission every 15 seconds
11	Counter	0 à 127	129	2nd Counter (7 Bit Counter with bit (Bits 0 à 6))
12	FW Version	0 à 255	31	Firmware Version Number (31/ 10) = v3.1
13	Reserved	ŠBŠ	66	Reserved
14	Reserved	ŠCŠ	67	Reserved
15	Alarm Counter	0 à 255	0	Movement Counter
16	Age byte	0 à 255	0	Life Cycle Counter Byte of Tag (MSB)
17	Age byte	0 à 255	152	Life Cycle Counter Byte of Tag
18	Age byte	0 à 255	136	Life Cycle Counter Byte of Tag
19	Age byte	0 à 255	60	Life Cycle Counter Byte of Tag (LSB)
20	Site code	0 à 255	0	Vendor Code (MSB)
21	Site code	0 à 255	0	Vendor Code
22	Site code	0 à 255	4	Vendor Code (LSB)
23	Tag ID	0 à 255	0	Tag ID (MSB)
24	Tag ID	0 à 255	0	Tag ID
25	Tag ID	0 à 255	0	Tag ID
26	Tag ID	0 à 255	108	Tag ID (LSB)
27	Type of tag	0 à 255	50	Tag is Fused. Cannot be re-programmed
		48 0x30		Tag is not Fused. Can be re-programmed.
		50 0x33		
28	Reader Address	0 à 255	80	Stored address of the reader
29	RSSI Value	0 à 255	127	RSSI Value (Signal Strength)
30	Tag CRC	0 à 255	177	Tag data checksum (see below to calculate)
31	Reserved	0x20	32	Reserved
32	Alarm byte	0 à 255	80	Alarm Byte (50 or 51)

Tabela F.3: Representação dos dados respondidos-continuação

Byte	Name	Data	Example	Description
33	Node ID	0 à 255	80	Node ID
34	Network ID	0 à 255	0	Network ID
35	Reader RSSI	0 à 255	0	RSSI Threshold set on the reader
36	FW Version	0 à 255	40	Firmware Version
37	LF	0 à 255	10	Post-amble
38	CR	0 à 255	13	Post-amble
39	Reader CRC	0 à 255	45	Checksum of total data packet

G. DESCRIÇÃO DO CONTEÚDO DO CD

O CD possui os *scripts* feitos no MatLab e no ActionView para a implementação deste trabalho, possui a pasta com os arquivos necessários para executar o módulo em tempo real do ActioView, e para a configuração do mesmo. Possui uma cópia do Trabalho de graduação e os arquivos latex.