

**UNIVERSIDADE DE BRASÍLIA**  
**FACULDADE DE TECNOLOGIA**  
**DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**REDES NEURAS ARTIFICIAIS APLICADAS AO  
PROBLEMA DA LOCALIZAÇÃO EM AMBIENTES  
FECHADOS**

**ROBERTO RODRIGUES LOIOLA**

**ORIENTADOR: ALEXANDRE RICARDO SOARES ROMARIZ**

**DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA**

**PUBLICAÇÃO: PPGENE.DM – 409/09**

**BRASÍLIA/DF: DEZEMBRO - 2009**

**UNIVERSIDADE DE BRASÍLIA**  
**FACULDADE DE TECNOLOGIA**  
**DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**REDES NEURAS ARTIFICIAIS APLICADAS AO  
PROBLEMA DA LOCALIZAÇÃO EM AMBIENTES  
FECHADOS**

**ROBERTO RODRIGUES LOIOLA**

DISSERTAÇÃO DE MESTRADO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA ELÉTRICA.

APROVADA POR:

---

**ALEXANDRE RICARDO SOARES ROMARIZ, Doutor, University of Colorado at Boulder, Estados Unidos (ORIENTADOR)**

---

**JANAÍNA GONÇALVES GUIMARÃES, Doutora, Universidade de Brasília, UnB, Brasília-DF (EXAMINADORA INTERNO)**

---

**MARCO ANTÔNIO ASSFALK DE OLIVEIRA, Doutor, University of New Mexico, UNM, Estados Unidos (EXAMINADOR EXTERNO)**

**DATA: BRASÍLIA/DF, 10 DE DEZEMBRO DE 2009.**







## FICHA CATALOGRÁFICA

LOIOLA, ROBERTO RODRIGUES

Redes Neurais Artificiais aplicadas ao problema da localização em ambientes fechados [Distrito Federal] 2009.

xxv, 171p., 210 x 297 mm (ENE/FT/UnB, Mestre, Dissertação de Mestrado – Universidade de Brasília. Faculdade de Tecnologia.

Departamento de Engenharia Elétrica

1.Localização em ambientes fechados

2.Redes Neurais Artificiais

3.Mapas auto-organizáveis de Kohonen

4.Computação Ubíqua

I. ENE/FT/UnB

II. Título (série)

## REFERÊNCIA BIBLIOGRÁFICA

LOIOLA, R. R. (2009). Redes Neurais Artificiais aplicadas ao problema da localização em ambientes fechados. Dissertação de Mestrado em Engenharia Elétrica, Publicação PPGENE.DM-409/09, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 171p.

## CESSÃO DE DIREITOS

AUTOR: Roberto Rodrigues Loiola.

TÍTULO: Redes Neurais Artificiais aplicadas ao problema da localização em ambientes fechados.

GRAU: Mestre

ANO: 2009

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

---

Roberto Rodrigues Loiola  
SQS 411 Bloco G Apartamento nº 310, Asa Sul.  
70277-070 Brasília – DF – Brasil.



## Dedicatória.

Dedico esta dissertação ao Senhor Raimundo e Dona Antônia, meus queridos pais,  
que ao longo de toda a minha caminhada de vida, não deixaram de acreditar  
em meu potencial e nunca negaram um conselho amigo,  
o esforço necessário e o amor incondicional.



## **AGRADECIMENTOS**

Em primeiro lugar, agradeço muito a Deus por me conceder a saúde, persistência e esperança necessárias para a conclusão de mais uma etapa de vida.

Ao Professor Alexandre Romariz, pela orientação objetiva, honesta e dedicada. Agradeço também a ele por ter me acolhido como orientando e acreditado em minha proposta de trabalho. A discussão aberta e clara dos rumos dessa pesquisa e a confiança no tema proposto foram essenciais para que as metas que traçamos fossem alcançadas.

Aos meus queridos pais, que desde muito cedo me ensinaram a importância dos estudos e do quanto se deve ser forte e perseverante nessa vida, nunca se deixando abater. Pai e Mãe, vocês nunca deixaram de acreditar em minha capacidade e fizeram todo o esforço para que eu sempre pudesse levar adiante meus estudos, até mesmo suportando os diversos momentos que eu passava no meu quarto nas madrugadas, atarefado e sem tempo, para que pudesse terminar tudo no prazo. Sem vocês eu jamais teria chegado a lugar algum. Agradeço profundamente o esforço, o amor e o carinho de vocês! Agradeço também ao meu irmão Ricardo, que durante todos os anos de vida tem sido um exemplo de determinação e persistência. Agradeço também a minha cunhada, Solange, por todo o incentivo e disposição em me auxiliar no que fosse necessário para que pudesse vencer diversas etapas deste trabalho.

Ao meu grande amigo Carlos Eduardo Neves, que nunca negou um conselho e ajuda sem tamanho nos momentos mais difíceis que passei. Não tenho dúvidas que ele é o amigo mais verdadeiro, fiel e dedicado que tenho o prazer de conviver há mais de uma década. Além disso, ele é um dos principais responsáveis por eu ter seguido com a carreira acadêmica.

Agradeço aos meus colegas e amigos da Engenharia Elétrica por serem exemplo de companheirismo, caráter e alegria dentro e fora da UnB. Em especial, aos meus amigos Alexandre Bellezi, Cláudio Monteiro, Edna Canedo, Guilherme Salazar e Mauro de Boni. A ajuda de vocês foi essencial para que pudesse chegar nessa fase.

A querida Josi Abreu, que nas horas mais difíceis dessa caminhada me deu o apoio, a compreensão e carinho necessários para essa reta final do mestrado.



## **RESUMO**

### **REDES NEURAIIS ARTIFICIAIS APLICADAS AO PROBLEMA DA LOCALIZAÇÃO EM AMBIENTES FECHADOS**

**Autor: Roberto Rodrigues Loiola**

**Orientador: Alexandre Ricardo Soares Romariz**

**Programa de Pós-Graduação em Engenharia Elétrica**

**Brasília, Dezembro de 2009**

Essa dissertação aborda o problema da localização em ambientes fechados baseada em técnicas de redes neurais artificiais. Nesse sistema, a informação da intensidade do sinal recebido (*RSSI*) disponibilizada por interfaces de rede sem fio padrão é a base para a previsão de localização de dispositivos móveis.

Métodos tradicionais de localização *indoor* possuem diversas características indesejáveis, tais como dificuldade de implementação, pouca flexibilidade (não permitem a utilização da infraestrutura presente no local sem grandes alterações na disposição de *APs*), número elevado de parâmetros e alto custo computacional.

Foram realizadas implementações de algoritmos tradicionais de localização (Algoritmo do Vizinho mais Próximo), métodos baseados em Redes Neurais Multicamadas (Perceptron MLP) e mapas auto-organizáveis de Kohonen. Conclui-se que esta última implementação (Kohonen) é capaz de prover resultados significativamente superiores àqueles obtidos em estudos recentes de localização *indoor*.





## **ABSTRACT**

### **ARTIFICIAL NEURAL NETWORKS APPLIED TO THE PROBLEM OF LOCATION IN INDOOR ENVIRONMENTS**

**Author: Roberto Rodrigues Loiola**

**Supervisor: Alexandre Ricardo Soares Romariz**

**Programa de Pós-Graduação em Engenharia Elétrica**

**Brasília, December of 2009**

This thesis addresses the indoor location problem using on artificial neural networks-based techniques. In this system, the received signal strength information (RSSI) provided by standard network wireless interfaces are the basis for mobile devices' location prediction.

Traditional methods of indoor location have several undesirable characteristics, such as implementation difficulties, lack of flexibility (requiring APs specific position), high number of parameters and high computational cost.

Traditional indoor location algorithms such as the Nearest Neighbor Algorithm were compared to methods based on Multilayer Neural Networks (Perceptron MLP) and the Kohonen self-organized map. We conclude that the Kohonen's implementation is able to provide significantly better results (less errors, faster localization) than those obtained in recent studies of indoor localization.



## SUMÁRIO

<b>1 - INTRODUÇÃO.....</b>	<b>1</b>
<b>1.1 - CONTRIBUIÇÕES DO TRABALHO.....</b>	<b>4</b>
<b>1.2 - ORGANIZAÇÃO DO TRABALHO.....</b>	<b>5</b>
<b>2 - FUNDAMENTOS TEÓRICOS (REVISÃO BIBLIOGRÁFICA) .....</b>	<b>6</b>
<b>2.1 - MÉTODOS CLÁSSICOS DE LOCALIZAÇÃO INDOOR .....</b>	<b>9</b>
<b>2.1.1 - Método Geométrico ou Método da Triangulação.....</b>	<b>9</b>
<b>2.1.2 - Método Probabilístico.....</b>	<b>11</b>
<b>2.1.3 - Algoritmo do Vizinho mais Próximo (<i>k</i>-Nearest Neighbor) .....</b>	<b>13</b>
2.1.3.1 - Conceitos Básicos sobre Aprendizado de Máquina.....	13
2.1.3.2 - Algoritmo <i>k</i> NN – <i>k</i> -Nearest Neighbor .....	14
<b>2.1.4 - Outros métodos clássicos de Localização Indoor .....</b>	<b>15</b>
<b>2.2 - MÉTODOS DE LOCALIZAÇÃO INDOOR BASEADOS EM REDES NEURAI</b>	
<b>ARTIFICIAIS.....</b>	<b>18</b>
<b>2.2.1 - As Redes Neurais Artificiais.....</b>	<b>18</b>
<b>2.2.2 - Redes Neurais Perceptron Multicamadas (MLP - Multilayered Perceptron)</b>	
.....	<b>18</b>
2.2.2.1 - Algoritmo de Retropropagação.....	19
<b>2.2.3 - Mapas Auto-organizáveis de Kohonen .....</b>	<b>21</b>
2.2.3.1 - Características básicas do modelo .....	22
2.2.3.2 - Etapas de Funcionamento do algoritmo .....	24
2.2.3.3 - Iterações.....	25
2.2.3.4 - O Algoritmo de Treinamento da Rede de Kohonen .....	25
2.2.3.5 - Atribuição de rótulos.....	26
<b>3 - AS REDES LOCAIS SEM FIO (WLANS) .....</b>	<b>28</b>
<b>3.1 – A FAMÍLIA 802.11.....</b>	<b>30</b>
<b>3.1.1 - O Indicador de Intensidade de Sinal (<i>RSSI – Received Signal Strength</i></b>	
<b><i>Indicator</i>) .....</b>	<b>30</b>
3.1.1.1 - Fatores que afetam o <i>RSSI</i> .....	33
3.1.1.2 - Aplicações do <i>RSSI</i> .....	36
<b>4 - METODOLOGIA EMPREGADA.....</b>	<b>39</b>
<b>4.1 - ESTIMAÇÃO DO MODELO .....</b>	<b>39</b>

4.1.1 - Aspectos computacionais .....	39
4.1.2 - Critérios para escolha e descrições do ambiente destinado aos testes experimentais .....	39
4.1.3 - Critérios para escolha dos <i>Access Points</i> – <i>APs</i> .....	41
4.1.4 - Aquisição experimental dos valores do sinal recebido ( <i>RSSI</i> ).....	45
4.1.5 - Análises das medidas experimentais e pré-processamento dos dados .....	47
<b>5 - RESULTADOS OBTIDOS.....</b>	<b>51</b>
<b>5.1 - REDE NEURAL DE KOHONEN.....</b>	<b>51</b>
<b>5.1.1 - Treinamento.....</b>	<b>51</b>
5.1.1.1 - Definição dos Parâmetros para Treino .....	51
5.1.1.2 - Análise preliminar (pré-treino) dos dados de entrada e possíveis agrupamentos.....	53
<b>5.1.2 - Resultados dos Treinamentos.....</b>	<b>59</b>
5.1.2.1 - Treinamento com a Planta Superior – SG11 .....	59
5.1.2.2 - Treinamento com a Planta Inferior – SG11.....	64
5.1.2.3 - Treinamento com a Planta Completa – SG11 .....	69
5.1.2.4 - Considerações a respeito da fase de Treinamento .....	72
<b>5.1.3 - Simulação.....</b>	<b>75</b>
5.1.3.1 - Simulação contendo pontos utilizados no treinamento.....	75
5.1.3.2 - Simulação contendo pontos localizados em regiões de fronteira .....	79
5.1.3.3 - Simulação contendo novos pontos adquiridos em novas medidas de campo aleatórias.....	84
<b>5.2 - REDE NEURAL PERCEPTRON MULTICAMADAS (MLP).....</b>	<b>86</b>
<b>5.2.1 - Definição das Variáveis de Treinamento .....</b>	<b>86</b>
<b>5.2.2 - Resultados dos Treinamentos.....</b>	<b>87</b>
5.2.2.1 - Treinamento com a Planta Superior – SG11 .....	88
5.2.2.2 - Treinamento com a Planta Inferior – SG11.....	90
5.2.2.3 - Treinamento com a Planta Completa – SG11 .....	91
5.2.2.4 - Considerações a respeito da fase de Treinamento .....	92
<b>5.2.3 - Simulação.....</b>	<b>94</b>
5.2.3.1 - Simulação contendo pontos utilizados no treinamento.....	95
5.2.3.2 - Simulação contendo pontos localizados em regiões de fronteira .....	99
5.2.3.3 - Simulação contendo novos pontos adquiridos em novas medidas de campo aleatórias.....	105

<b>5.3 - ALGORITMO DO VIZINHO MAIS PRÓXIMO (NEAREST NEIGHBOR - NN)</b>	<b>112</b>
.....	112
<b>5.3.1 - Definição das Variáveis do Algoritmo NN</b>	<b>112</b>
<b>5.3.2 - Simulações</b>	<b>113</b>
5.3.2.1 - Simulação contendo pontos utilizados no treinamento e pontos localizados em regiões de fronteira	113
5.3.2.2 - Simulação contendo novos pontos adquiridos em novas medidas de campo aleatórias	113
<b>5.4 - ANÁLISES COMPARATIVAS: REDES NEURAIAS DE KOHONEN, PERCEPTRON MULTICAMADAS (MLP) E ALGORITMO DO VIZINHO MAIS PRÓXIMO (NEAREST NEIGHBOR - NN)</b>	<b>116</b>
<b>5.4.1 - Simulação contendo 75% dos pontos originais</b>	<b>117</b>
<b>5.4.2 - Simulação para as Regiões de Fronteira</b>	<b>119</b>
<b>5.4.3 - Simulação para novas medidas experimentais</b>	<b>121</b>
<b>5.4.4 - Total de Iterações na fase de Treinamento</b>	<b>122</b>
<b>5.4.5 - Tempo de Execução dos Algoritmos de Treinamento</b>	<b>124</b>
<b>5.4.6 - Tempo de Execução dos Algoritmos baseado na quantidade de pedidos de localização</b>	<b>126</b>
<b>6 - CONCLUSÕES</b>	<b>131</b>
<b>6.1 SUGESTÕES PARA PESQUISAS FUTURAS</b>	<b>133</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS</b>	<b>135</b>
<b>APÊNDICES</b>	<b>144</b>
<b>A – DESCRIÇÃO DOS APS UTILIZADOS NO EXPERIMENTO</b>	<b>145</b>

## LISTA DE TABELAS

Tabela 4.1 - Quadro resumo contendo os aspectos computacionais empregados no trabalho.	39
Tabela 4.2 - Quadro resumo contendo a identificação e localização dos APs utilizados no experimento.....	42
Tabela 4.3 - Quadro resumo das medidas experimentais .....	45
Tabela 4.4 - Distribuição dos Pontos entre Sobreloja e Térreo – SG 11 .....	46
Tabela 4.5 - Quadro resumo das medidas experimentais expressas em dBm .....	47
Tabela 4.6 - Análise dos dados obtidos para o ambiente “Gerência” .....	48
Tabela 4.7 - Dados após a aproximação, para o ambiente “Gerência” .....	49
Tabela 4.8 - Quadro resumo das medidas experimentais expressas em dBm após filtragem dos erros de medida.....	50
Tabela 5.1 - Parâmetros utilizados por tipo de treinamento para a RNA Kohonen.....	74
Tabela 5.2 - Tempo de Execução na fase de Treinamento para a RNA Kohonen (no formato: h:min:s) .....	74
Tabela 5.3 - Parâmetros de divisão de amostras para treinamento para Rede MLP (Plantas Superior, Inferior e Completa) .....	88
Tabela 5.4 - Resultados do Treinamento para Planta Superior – Rede Neural MLP.....	93
Tabela 5.5 - Resultados do Treinamento para Planta Inferior – Rede Neural MLP .....	94
Tabela 5.6 - Resultados do Treinamento para Planta Completa – Rede Neural MLP.....	94
Tabela 5.7 – Erro médio em metros para Pontos Conhecidos - Planta Superior – Rede MLP	96
Tabela 5.8 - Erro médio em metros para Pontos Conhecidos - Planta Inferior – Rede MLP..	97
Tabela 5.9 - Erro médio em metros para Pontos Conhecidos - Planta Completa – Rede MLP	98
Tabela 5.10 - Erro médio em metros para Regiões de Fronteira - Planta Superior – Rede MLP .....	102
Tabela 5.11 - Erro médio em metros para Regiões de Fronteira - Planta Inferior – Rede MLP .....	103
Tabela 5.12 - Erro médio em metros para Regiões de Fronteira - Planta Completa – Rede MLP .....	104
Tabela 5.13 - Erro médio em metros para Novos Pontos - Planta Superior – Rede MLP .....	106
Tabela 5.14 - Erro médio em metros para Novos Pontos - Planta Inferior – Rede MLP.....	107
Tabela 5.15 - Erro médio em metros para Novos Pontos - Planta Completa – Rede MLP....	108
Tabela 5.16 - Tempo de Execução das Simulações – Algoritmo NN, Redes Neurais MLP e Kohonen .....	129

## LISTA DE FIGURAS

Figura 2.1 - Relação entre a intensidade do sinal (expressa em dB) e a distância (em metros) [65].....	7
Figura 2.2 – Interpretação geométrica da triangulação [62].....	10
Figura 2.3 – Exemplo de classificação do método <i>k-Nearest Neighbor</i> [26].....	15
Figura 2.5 – Mapas auto-organizáveis [30].....	23
Figura 3.1 – Formato PLCP PPDU [34].....	31
Figura 3.2 - Arquitetura de um circuito para medição do <i>RSSI</i> [33].....	33
Figura 3.3 - Exemplo do padrão de radiação de uma antena omni-direcional [51].....	35
Figura 4.1 - Localização dos <i>APs</i> SG-11 – Térreo– Escala 1/100.....	43
Figura 4.2 - Localização dos <i>APs</i> SG-11 – Sobreloja – Escala 1/100.....	44
Figura 4.3 - Software de medição inSSIDer contendo todos os <i>APs</i> encontrados no SG-11...46	
Figura 4.4 - Etapas da medição utilizando o inSSIDer para os <i>APs</i> de referência. Todas as medidas foram realizadas com as portas fechadas e sem pessoas transitando ao longo dos corredores/laboratórios. ....	47
Figura 5.1 - Ponto de Acesso “WLPCI” (localizado no térreo)– Planta Superior SG-11 .....	54
Figura 5.2 - Ponto de Acesso “LabFontes”(localizado no térreo) – Planta Superior SG-11....	54
Figura 5.3 - Ponto de Acesso “GRAV”(localizado no piso superior) – Planta Inferior SG-11	55
Figura 5.4 - Análise das Medidas - Ponto de Acesso “LAVSI” (localizado no piso superior)– Planta Inferior SG-11.....	55
Figura 5.5 - Emissor WLPCI - Localizado no mesmo andar – Planta Inferior .....	57
Figura 5.6 - Emissor LabFontes – Localizado no mesmo andar – Planta Inferior .....	57
Figura 5.7 - Emissor GRAV – Localizado no mesmo andar – Planta Superior .....	58
Figura 5.8 - Emissor LAVSI – Localizado no mesmo andar – Planta Superior.....	58
Figura 5.9 – Consolidação dos Agrupamentos em 500 iterações dos Padrões encontrados na Planta Superior – SG11.....	60
Figura 5.10 - Consolidação dos Agrupamentos em 1.000 iterações dos Padrões encontrados na Planta Superior – SG11.....	61
Figura 5.11 - Total de Pontos pelo Total de Padrões encontrados – Planta Superior – 5.000 iterações .....	62
Figura 5.12 - Consolidação dos Agrupamentos dos Padrões encontrados em 5.000 iterações na Planta Superior – SG11.....	63

Figura 5.13 - Consolidação dos Agrupamentos em 500 iterações dos Padrões encontrados na Planta Inferior – SG11 .....	65
Figura 5.14 - Consolidação dos Agrupamentos em 1.000 iterações dos Padrões encontrados na Planta Inferior – SG11 .....	66
Figura 5.15 - Total de Pontos pelo Total de Padrões encontrados – Planta Inferior – 5.000 iterações .....	67
Figura 5.16 - Consolidação dos Agrupamentos dos Padrões encontrados em 5.000 iterações na Planta Inferior – SG11 .....	68
Figura 5.17 - Mapa Resultante do Treinamento da Rede Neural de Kohonen – Planta Completa – 10.000 iterações .....	69
Figura 5.18 - Consolidação dos Agrupamentos dos Padrões encontrados em 10.000 iterações na Planta Completa (Superior) – SG11 .....	70
Figura 5.19 - Consolidação dos Agrupamentos dos Padrões encontrados em 10.000 iterações na Planta Completa (Inferior)– SG11 .....	71
Figura 5.20 - Decaimento exponencial da função de distância de vizinhança – Planta Completa – SG-11 .....	73
Figura 5.21 - Decaimento exponencial da taxa de aprendizado – Planta Completa – SG-11 ..	73
Figura 5.22 - Seleção aleatória de 75% das amostras para a Planta Superior – RNA Kohonen .....	76
Figura 5.23 - Seleção aleatória de 75% das amostras para a Planta Inferior – RNA Kohonen	77
Figura 5.24 - Resultados alcançados para a Planta Superior – Quantitativos de Pontos – RNA Kohonen – 100% de Acerto .....	78
Figura 5.25 - Resultados alcançados para a Planta Inferior – Quantitativos de Pontos – RNA Kohonen – 97,78% de Acerto e 2,22% de Erro .....	78
Figura 5.26 - Resultados alcançados para a Planta Completa (Resultados Consolidados) – Quantitativos de Pontos – RNA Kohonen – 98,99% de Acerto e 1,01% de Erro.....	79
Figura 5.27 - Pontos selecionados em Regiões de Fronteira – Planta Superior – RNA Kohonen .....	80
Figura 5.28 - Pontos selecionados em Regiões de Fronteira – Planta Inferior – RNA Kohonen .....	81
Figura 5.29 - Simulação para Pontos das Fronteiras das Classes - Somente Planta Superior - Quantitativo de Pontos – RNA Kohonen – 100% de Acerto .....	82
Figura 5.30 - Simulação para Pontos das Fronteiras das Classes - Somente Planta Inferior - Quantitativo de Pontos – RNA Kohonen – 100% de Acerto .....	82



Figura 5.31 - Simulação para Pontos das Fronteiras das Classes - Planta Completa (Consolidada) - Quantitativo de Pontos – RNA Kohonen – 100% de Acerto .....	83
Figura 5.32 - Simulação para Novos Pontos - Resultado da Planta Completa – Total de Pontos – RNA Kohonen – 99,10% de Acerto e 0,9% de Erro .....	84
Figura 5.33 - Simulação para Novos Pontos – Somente Planta Superior – Total de Pontos – RNA Kohonen – 99,34% de Acerto e 0,66% de Erro .....	85
Figura 5.34 - Simulação para Novos Pontos – Somente Planta Inferior – Total de Pontos – RNA Kohonen – 98,80% de Acerto e 1,2% de Erro .....	85
Figura 5.35 - Treinamento da Planta Superior com 150 neurônios nas camadas intermediárias (Melhor configuração) – Rede MLP.....	89
Figura 5.36 - Treinamento da Planta Superior com 150 neurônios nas camadas intermediárias (Melhor Resultado da Validação) – Rede MLP .....	89
Figura 5.37 - Treinamento da Planta Inferior com 200 neurônios nas camadas intermediárias (Melhor configuração) – Rede MLP.....	90
Figura 5.38 - Treinamento da Planta Inferior com 200 neurônios nas camadas intermediárias (Melhor Resultado da Validação) – Rede MLP .....	91
Figura 5.39 - Treinamento da Planta Completa com 250 neurônios nas camadas intermediárias (Melhor configuração) – Rede MLP.....	92
Figura 5.40 - Treinamento da Planta Completa com 250 neurônios nas camadas intermediárias (Melhor Resultado da Validação) – Rede MLP .....	92
Figura 5.41 - Resultados alcançados para a Planta Superior – Quantitativos de Pontos – Rede MLP – 96,27% de Acerto e 3,73% de Erro.....	96
Figura 5.42 - Resultados alcançados para a Planta Inferior – Quantitativos de Pontos – Rede MLP – 98,10% de Acerto e 1,9% de Erro .....	97
Figura 5.43 - Resultados alcançados para a Planta Completa – Quantitativos de Pontos – Rede MLP – 96,81% de Acerto e 3,19% de Erro.....	98
Figura 5.44 - Localização dos novos Pontos para as regiões de fronteira – Planta Superior – Rede MLP .....	100
Figura 5.45 - Localização dos novos Pontos para as regiões de fronteira – Planta Inferior – Rede MLP .....	101
Figura 5.46 - Resultados alcançados para regiões de fronteira - Planta Superior – Quantitativo de Pontos – Rede MLP – 87,56% de Acerto e 12,44% de Erro.....	102
Figura 5.47 - Resultados alcançados para regiões de fronteira - Planta Inferior – Quantitativo de Pontos – Rede MLP – 89,77% de Acerto e 10,23% de Erro.....	103

Figura 5.48 - Resultados alcançados para regiões de fronteira - Planta Completa – Quantitativo de Pontos – Rede MLP – 88,64% de Acerto e 11,36% de Erro.....	104
Figura 5.49 - Resultados alcançados para novos pontos - Planta Superior – Quantitativo de Pontos – Rede MLP – 88,52% de Acerto e 11,48% de Erro .....	106
Figura 5.50 - Resultados alcançados para novos pontos - Planta Inferior – Quantitativo de Pontos – Rede MLP – 92% de Acerto e 8 % de Erro .....	107
Figura 5.51 - Resultados alcançados para novos pontos - Planta Completa – Quantitativo de Pontos – Rede MLP – 90,09% de Acerto e 9,91% de Erro .....	108
Figura 5.52 - Localização Original VS Localização Estimada pela RNA MLP – Planta Superior – 35 Medidas – Rede MLP .....	110
Figura 5.53 - Localização Original VS Localização Estimada pela RNA MLP – Planta Inferior – 20 Medidas – Rede MLP .....	111
Figura 5.54 - Resultados alcançados para novos pontos - Planta Superior – Quantitativo de Pontos – NN – 85,25% de Acerto e 14,75 % de Erro.....	114
Figura 5.55 - Resultados alcançados para novos pontos - Planta Inferior – Quantitativo de Pontos – NN – 54% de Acerto e 46% de Erro .....	114
Figura 5.56 - Resultados alcançados para novos pontos - Planta Completa – Quantitativo de Pontos – NN – 71,17% de Acerto e 28,83% de Erro.....	115
Figura 5.57 - Comparação do resultado da Simulação para Pontos conhecidos – Redes Kohonen, MLP e Algoritmo NN .....	117
Figura 5.58 - Comparação do resultado percentual da Simulação para Pontos Conhecidos – Redes Kohonen, MLP e Algoritmo NN.....	118
Figura 5.59 - Comparação do resultado da Simulação em Regiões de Fronteira – Redes Kohonen, MLP e Algoritmo NN .....	120
Figura 5.60 - Comparação do resultado percentual da Simulação em Regiões de Fronteira – Redes Kohonen, MLP e NN.....	120
Figura 5.61 - Comparação do resultado da Simulação para novas medidas – Redes Kohonen, MLP e Algoritmo NN.....	121
Figura 5.62 - Comparação do percentual do resultado da Simulação para Novas Medidas – Redes Kohonen, MLP e Algoritmo NN.....	122
Figura 5.63 - Comparativo do Total de Iterações do Algoritmo de Treinamento – Redes Kohonen, MLP e Algoritmo NN .....	123
Figura 5.64 - Comparativo do Tempo de Execução do Algoritmo de Treinamento das Redes Neurais Kohonen, MLP e Algoritmo NN .....	125

Figura 5.65 - Tempo de Execução da Simulação por Quantidade de Medidas – Algoritmo NN, Redes Neurais MLP e Kohonen .....	127
Figura 5.66 - Tempo de Execução da Simulação por Quantidade de Medidas –Redes Neurais MLP e Kohonen .....	128

## LISTA DE SÍMBOLOS, NOMENCLATURA E ABREVIACÕES

ADC	Conversor Analógico-Digital
APs	<i>Access Points</i>
CTS	<i>Clear to Send</i>
CSMA/CA	<i>Carrier-Sense Multiple Access, Collision Avoidance</i>
dBm	<i>Power Ratio in Decibels</i>
DSSS	<i>Direct Sequence Spread Spectrum</i>
FCC	<i>Federal Communications Commission</i>
FHSS	<i>Frequency Hopping Spread Spectrum</i>
FLOPS	<i>Floating point Operations Per Second</i>
GPS	<i>Global Positioning System</i>
IEEE	<i>Institute of Electrical and Electronic Engineers</i>
kNN	<i>k-Nearest Neighbor</i>
LBS	<i>Location Based Service</i>
MAC	<i>Media Access Control</i>
MLP	<i>Multi-layer Perceptron</i>
NIC	<i>Network Interface Card</i>
NN	<i>Nearest Neighbor</i>
OFDM	<i>Orthogonal Frequency-Division Multiplexing</i>
PHY	<i>Physical Layers</i>
PLCP	<i>Higher Physical Layer Convergence Protocol</i>
PMD	<i>Lower Physical Media Dependent</i>
RF	Rádio-frequência
RFID	<i>Radio-Frequency Identification</i>

RNA	Rede Neural Artificial
RSS	<i>Radio Spread Spectrum</i>
RSSI	<i>Received Signal Strength Indicator</i>
SIG	<i>Sistemas de Informação Geográfica</i>
WLAN	<i>Wireless Local Area Network</i>

# 1 - INTRODUÇÃO

À medida que a computação ubíqua torna-se mais popular, a necessidade de aplicativos cientes do contexto aumenta. O contexto de uma aplicação é relacionado com a informação que é parte do seu ambiente operacional. Normalmente, esse contexto inclui informações como localização, atividade de pessoas e o estado de outros dispositivos [29]. Algoritmos e técnicas que permitam que uma aplicação esteja ciente da localização de um dispositivo em um ambiente mapeado é um pré-requisito para muitas destas aplicações [17,72,53,55,27]

A crescente necessidade de sistemas que auxiliem na localização ressalta a importância de se investigarem técnicas eficientes para o problema. Como exemplo, podemos citar as iniciativas governamentais (principalmente na América do Norte) que exigem dos provedores de telefonia móvel uma maneira de localizar qualquer dispositivo móvel no instante em que realiza uma chamada de emergência [25]. Em sistemas *outdoor*, que são aqueles localizados no espaço-livre, sistemas de posicionamento global (GPS) [22] têm sido utilizados em muitas aplicações comerciais, como no caso de localização de automóveis e cargas.

O sucesso do GPS inspirou o surgimento de vários sistemas de localização. Alguns deles são sistemas de navegação global por satélite (GNSS), e são concorrentes do sistema GPS, como o russo *GLONASS* e o sistema ainda em fase de testes, da União Europeia, *Galileo*. Alguns sistemas de localização são concebidos como um complemento ao sistema GPS, tais como o *GpsOne* da *QUALCOMM*, A-GPS ou tecnologias baseadas em sistemas de telecomunicações móveis. O *GpsOne* utiliza uma combinação de sinais de satélite GPS e torres de sinais de celular que permite a localização do usuário com maior precisão que os tradicionais sistemas de GPS em áreas onde a recepção por satélite é problemática devido a edifícios ou relevos.

Entretanto, muitos ambientes fechados (*indoor*) não conseguem receber, com fidelidade e precisão, os sinais GPS. Em ambientes com essas características, pode-se empregar diferentes sensores e técnicas, como o infravermelho [71,6], visão computacional

[42,77,5], contato físico [52], ultrassom [73,57,31] ou rádio-frequência (RF) [1-28;43;59;60-88].

Além de um serviço de posicionamento, sistemas de localização também podem trazer muitas outras aplicações. Para a indústria de comunicação móvel, que atualmente está imersa em forte concorrência, as operadoras procuram constantemente novas maneiras para criar diferenciação no mercado e aumentar os lucros. Uma das melhores maneiras de conseguir isso é através da prestação de serviços altamente personalizados. Uma das mais poderosas maneiras de personalizar os serviços móveis é baseada em localização. *Location Based Service* (LBS) é considerado como fonte potencial de lucro no futuro próximo. A tecnologia LBS faz uso de Localização, Sistemas de Informação Geográfica (SIG), e de funcionalidades de gerenciamento de localização com o intuito de fornecer aos utilizadores de serviços, tais como a fornecimento de informações de acordo com a localização, faturamento de serviços por localização, os serviços de emergência, monitoramento, entre outros. Nos Estados Unidos, a FCC (*Federal Communications Commission*) - agência governamental que regula o uso das emissões de radiofrequência (RF) - exige que todas as operadoras de Telecom cumpram certos critérios para respeitar a LBS (FCC 94-102). A exigência prevê 95% de precisão de 300 metros para aparelhos com localização baseada na rede e 150 metros para telefones com rastreamento nativos. Isso pode ser especialmente útil quando o usuário discar para um número de telefone de emergência, de modo que o operador possa enviar os serviços de emergência, para o local correto.

A classe de sistemas baseada em comunicações via rádio-frequência (RF) que utilizam uma rede de dados sem fio [70-80], tais como a rede sem fio padrão IEEE 802.11 [35], para estimar a localização de usuários ou terminais móveis tem ganhado, gradativamente, diversos estudos, especialmente para casos de ambientes fechados. Muitos dos dispositivos móveis e também edifícios, tanto comerciais como residenciais, já estão equipados com APs sem fio padrão IEEE 802.11, os *Access Points* (APs). Além disso, a maioria dos dispositivos de acesso a redes sem fio são capazes de medir a intensidade do sinal recebido como parte do processo padrão de operação e é possível observar também como a intensidade do sinal varia com a distância e obstruções entre as alterações dos nós da rede.

Redes *WLAN* são vantajosas em termos de mobilidade, flexibilidade, área de cobertura de banda larga e facilidade de configuração. Todos esses fatores possibilitam o desenvolvimento de novas aplicações. Considerando que redes *WLAN* são normalmente instaladas em ambientes fechados, elas fornecem plataformas ideais para sistemas de localização *indoor*.

Uma importante aplicação da tecnologia *WLAN* engloba a computação baseada em contexto. Sistemas baseados em contexto referem-se a sistemas que podem ter o conhecimento de suas características físicas e virtuais no ambiente ou situação a que estejam submetidos e responder inteligentemente com base em tais informações.

Computação ubíqua integra a computação no ambiente. Informações de localização provêm contexto adicional para localização de estações móveis ubíquas e, portanto, o desenvolvimento de técnicas de localização baseadas em redes *WLANs* é relevante para aplicações baseadas em contexto e computação ubíqua. Essa técnica poderá também beneficiar indústrias e serviços de emergência que requerem a localização e o rastreamento de objetos e pessoas em ambientes fechados.

Exemplos de aplicações incluem monitoramento de bens e produtos dentro de uma fábrica, a localização de equipes médicas e pacientes nos hospitais, e monitoramento de emergências. Algumas outras aplicações que podem ser exploradas com essa técnica e discussões a respeito da computação ubíqua serão apresentadas oportunamente.

Seguindo essa linha, se um sistema de localização com precisão adequada puder ser desenvolvido utilizando somente esta tecnologia (*WLAN*), muitos dos sistemas existentes poderão ser reconvertidos via software e novos sistemas poderão ser implantados facilmente utilizando tecnologia já estabelecida e disponível.

Diante do exposto, essa dissertação aborda o problema da localização em ambientes fechados baseada em técnicas de redes neurais artificiais. Nesse sistema, a informação da intensidade do sinal recebido (*RSSI*) disponibilizada por interfaces de rede sem fio padrão é a base para a previsão de localização de dispositivos móveis.



Métodos tradicionais de localização *indoor*, utilizados até os dias atuais, possuem diversas características negativas que impossibilitam a adoção em massa do padrão, tais como dificuldade de implementação, poucas referências detalhadas a respeito do funcionamento dos algoritmos, pouca ou quase nenhuma flexibilidade (não permitem a utilização da infra-estrutura presente no local sem grandes alterações na disposição de APs), número elevado de parâmetros, alto custo computacional nem sempre associado com uma precisão que os justifique, além de diversas outras características indesejáveis.

## 1.1 - CONTRIBUIÇÕES DO TRABALHO

Esta dissertação traz a avaliação do desempenho de um mapa auto-organizável de Kohonen que utiliza a infra-estrutura de redes sem fio de um ambiente fechado para a localização de dispositivos móveis. O estudo leva em consideração o valor da intensidade do sinal recebido (*RSSI – Received Signal Strength*) de APs de referência e os compara com métodos tradicionais de localização *indoor* (Algoritmo do Vizinho mais Próximo) e a técnica de localização baseada em Redes Neurais Artificiais do tipo Perceptron Multicamadas – MLP.

Objetiva-se avaliar o potencial da utilização de uma abordagem baseada na rede neural de Kohonen e compará-la com a abordagem baseada em Rede Neural MLP e Algoritmo do Vizinho mais Próximo em termos de tempo de execução dos algoritmos de treinamento, capacidade de processamento das requisições de localização, precisão na localização de pontos conhecidos (mensurados experimentalmente), localizados em regiões entre ambientes e novas medidas adquiridas no ambiente em estudo.

Os resultados demonstram um incremento na capacidade de localização de dispositivos no ambiente selecionado quando utilizado o mapa de Kohonen. Os estudos mostraram aumentos expressivos na precisão da localização dos pontos chegando a um aumento de precisão de 11x comparado com a Rede Neural Multicamadas, redução do tempo de processamento de medidas (comparado com as Redes Neurais MLP e Algoritmo NN), redução no custo de implementação de um sistema de localização (o projeto é adaptável a infra-estrutura de redes sem fio do ambiente) e possibilidade de processamento paralelo para redução do tempo de resposta as localizações e aumento do número de consultas simultâneas direcionadas a Rede Neural. Tais fatos reforçam a necessidade de

abordagens mais avançadas baseadas em modelos neurais, em especial, as Redes Neurais de Kohonen.

## **1.2 - ORGANIZAÇÃO DO TRABALHO**

Este trabalho está dividido em 6 capítulos e um apêndice.

No presente capítulo são apresentados a relevância da pesquisa, objetivos e um resumo da metodologia empregada na mesma.

No **Capítulo 2** será apresentada uma breve revisão bibliográfica sobre os principais métodos de localização *indoor* e os métodos mais atuais baseados em redes neurais artificiais (Redes Neurais Perceptron Multicamadas e Mapas Auto-organizáveis de Kohonen).

O **Capítulo 3** apresenta detalhes sobre as redes locais sem fio (WLANs), em especial, o *RSSI* que é a base para a localização que utiliza as redes neurais artificiais. O capítulo também relaciona as Redes Neurais com as Redes WLANs em termos de aplicações do *RSSI*, além dos fatores que afetam a intensidade do sinal sem fio.

No **Capítulo 4** será discutida a metodologia empregada para a estimação do modelo utilizado nessa dissertação.

No **Capítulo 5** serão apresentados os resultados dos algoritmos de localização com as respectivas análises.

No **Capítulo 6** serão apresentadas as conclusões e sugestões para pesquisas futuras.

No Apêndice A encontram-se as especificações técnicas dos *Access Points* que foram utilizados para a simulação dos problemas apresentados.

## 2 - FUNDAMENTOS TEÓRICOS (REVISÃO BIBLIOGRÁFICA)

Utilizar uma rede sem fio para a determinação da localização de dispositivos é um grande desafio, uma vez que o sistema deve conseguir lidar com as características de ruído presentes nesse tipo de canal de comunicação. O padrão IEEE 802.11 utiliza a radiofrequência de comunicação na banda de 2,4 GHz, o que é muito interessante, pois essa faixa é isenta de licença de utilização para a maioria dos lugares do mundo. Os adaptadores atualmente disponíveis utilizam a tecnologia de rádio conhecida como espalhamento de espectro (*RSS - Radio Spread Spectrum*), na qual o sinal é espalhado em frequência [65], assim, eventuais interferências em uma frequência não bloqueiam todo o sinal. O principal problema é que prever com precisão a intensidade do sinal em qualquer posição do ambiente é uma tarefa difícil, porque o sinal propaga-se em muitas formas [70]. O sinal recebido ainda pode ser corrompido por efeitos indesejados, tais como interferências de outras fontes e interferências entre canais.

À medida que as ondas se propagam ao longo do ambiente, esse mesmo ambiente espalha as ondas de diferentes formas. Reflexão, absorção e difração ocorrem quando as ondas encontram obstáculos opacos. Refração ocorre quando as ondas encontram obstáculos translúcidos. Alterações nas condições atmosféricas, como temperatura do ar, podem afetar também a propagação das ondas e as intensidades dos sinais resultantes. Infelizmente, a frequência de 2,4 GHz é uma frequência ressonante na água, assim, pessoas também absorvem ondas de rádio na frequência de 2,4 GHz.

A interferência ocorre quando outra fonte gera um sinal na mesma frequência com intensidade significativa comparada à do sinal transmitido. O dispositivo que causa a interferência não precisa ser um dispositivo que transmite nas ondas de rádio [65, 59, 81]. Na faixa de frequência de 2,4 GHz, fornos microondas, dispositivos Bluetooth, telefones sem fio 2,4 GHz e equipamento de solda podem ser fontes de interferência. Devido à reflexão, refração e difração das ondas de rádio por estruturas e pessoas dentro de um edifício, o sinal transmitido frequentemente atinge o receptor por mais de um caminho, resultando em um fenômeno conhecido como atenuação multi-percurso [46].

Os componentes do sinal chegam ao receptor por caminhos diretos e indiretos, combinam-se e produzem uma versão distorcida do sinal transmitido. O desvanecimento multi-percurso é a principal causa das variações em pequena escala em que uma pequena mudança na posição do receptor (ordem de comprimentos de onda) podem conduzir a uma mudança significativa no sinal recebido [65].

Estas dificuldades são particularmente graves quando se trabalha em ambientes fechados. Uma vez que, raramente existe linha de visada entre transmissor e receptor, o sinal recebido é uma soma de componentes que são freqüentemente causados por uma combinação dos fenômenos descritos anteriormente. O sinal recebido varia com relação ao tempo e, especialmente no que diz respeito à posição relativa do receptor e do transmissor.

As técnicas atuais de determinação da localização de dispositivos em redes sem fio padrão 802.11 sofrem com o fenômeno do ruído, levando a uma precisão comprometida. A figura 2.1 mostra a relação entre a intensidade do sinal e a distância em uma área de 12 x 21 metros quadrados.

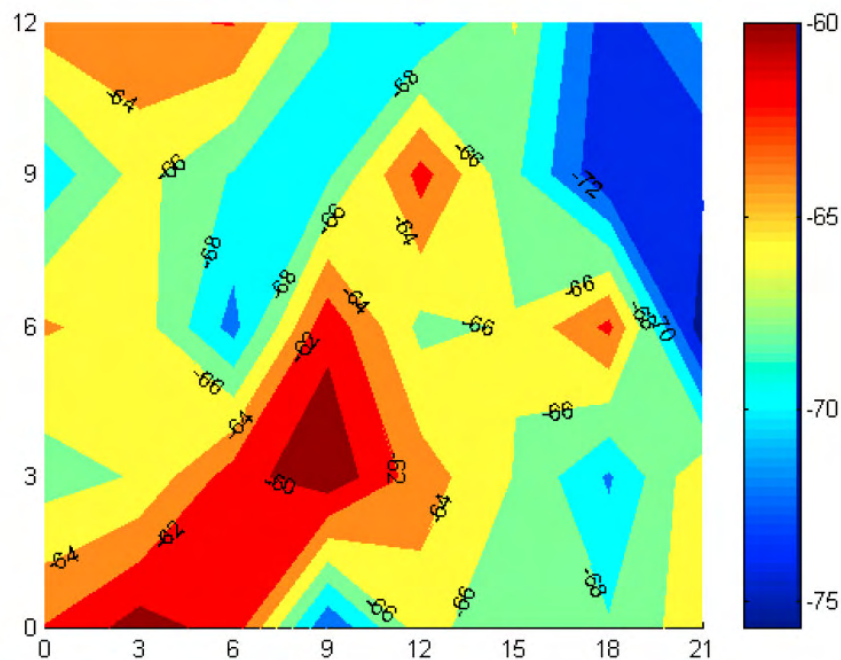


Figura 2.1 - Relação entre a intensidade do sinal (expressa em dB) e a distância (em metros) [65]

Como resultado do canal com ruído, é difícil entender a relação entre a intensidade do sinal e a distância, em um ambiente doméstico, com uma simples função analítica. Em vez disso, a localização em ambientes fechados baseada na tecnologia WLAN pode capturar informações relevantes de diferentes fontes em locais selecionados, dentro de uma determinada área de interesse. Os agrupamentos dessas informações constituirão como indica a literatura, um Mapa de Propagação. Portanto, esse mapa, que ajudará na determinação da localização, trabalha em duas fases: Uma fase *offline*, em que o mapa é propriamente constituído, e a fase *online*, onde amostras de intensidade do sinal recebidas de *APs* são utilizadas para "procurar" no mapa criado a localização estimada do usuário.

O sistema de estimação da localização pode ser implementado de modo centralizado ou em cada dispositivo móvel. Uma aplicação centralizada tem a vantagem de reduzir a demanda computacional no dispositivo móvel. Além disso, como dispositivos móveis apresentam problemas de autonomia de bateria é importante que se reduza o custo computacional para esses sistemas de localização.

Com isso, um sistema de localização deve cumprir alguns pré-requisitos:

- **Alta precisão:** Este é o objetivo principal de qualquer sistema de localização. Embora a precisão desejada seja dependente da aplicação em uso, quanto maior a precisão, maior o leque de aplicações que esse sistema conseguirá suportar.
- **Baixo *overhead* computacional:** Alta precisão não deve criar um alto custo computacional, especialmente se a determinação da localização for embutida no dispositivo móvel e não em um modo centralizado (proposta do presente trabalho).
- **Desenho flexível:** Tipicamente, é desejável que um sistema de localização possa ser desenvolvido em diferentes arquiteturas de hardware. A concepção de flexibilidade permite que o sistema possa ser portado entre várias arquiteturas com mínimas alterações.

- **Escalabilidade:** O sistema deve suportar localização em grandes áreas e um grande número de usuários.

Métodos de propagação baseados em padrões, também chamados de *fingerprinting* são considerados, nos últimos tempos, como os principais métodos de localização *indoor*. O método de *fingerprinting* é baseado na teoria de que a informação sobre localização pode ser extraída ou estimada através da comparação da característica do sinal detectado com as características pré-armazenadas do sinal. Em uma rede WLAN, esta técnica requer um mapa de propagação que é formado por vetores contendo a intensidade do sinal recebido (*RSS*) a partir de um conjunto de *APs* para locais específicos, e cada vetor do *RSS* é único para um local específico. Quando um dispositivo móvel precisa estimar a sua localização, ele mede o *RSSI* de diferentes *APs* para sua localização atual e pesquisa o sinal padrão com a correspondência mais próxima no mapa de propagação para extrair a localização correspondente [54].

A aplicação do método de localização em ambientes fechados baseada em redes WLAN tem atraído diversos pesquisadores com interesse no assunto. Diferentes métodos de extração da localização em ambientes fechados a partir de um método de propagação foram propostos, tais como o método do Vizinho mais próximo (*RADAR System*) [7], o método probabilístico [59,81,60,14], e o método baseado em Redes Neurais Artificiais [9,11]. Alguns pesquisadores [37,41,54,74,21] compararam os métodos acima, criando e realizando experiências em seu próprio ambiente de testes. Um método que vem se destacando com vantagem sobre todos eles é aquele baseado em Redes Neurais Artificiais, [9,11] foco de estudo deste trabalho.

## **2.1 - MÉTODOS CLÁSSICOS DE LOCALIZAÇÃO INDOOR**

### **2.1.1 - Método Geométrico ou Método da Triangulação**

Em sua forma mais geral de interpretação, triangulação é uma técnica geométrica que utiliza intervalos ou intersecções entre objetos para determinar a posição [62]. Um objeto é considerado posicionado de forma única quando ao menos três pontos de

referência são associados em um espaço bi-dimensional, formando um triângulo. Esses laços entre os objetos possibilitam a medição de distâncias e ângulos (figura 2.2).

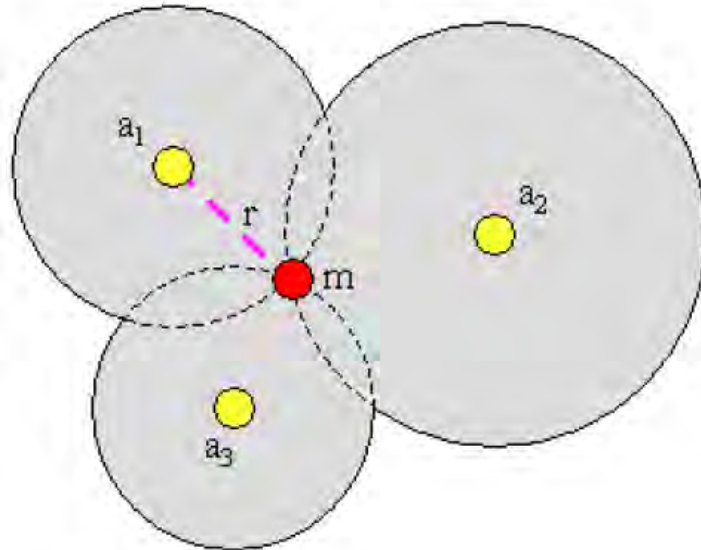


Figura 2.2 – Interpretação geométrica da triangulação [62]

Considere um espaço bi-dimensional com um objeto,  $m$ , em uma posição não identificada, alcançado por três pontos de referência,  $a_1$ ,  $a_2$ , e  $a_3$ , todos localizados em posições conhecidas. Uma vez que  $m$  determina a magnitude da distância,  $r$ , de  $a_1$ ,  $m$  pode inferir que está localizado em uma região do perímetro de um círculo centrado em  $a_1$  com raio  $r$ . O mesmo raciocínio a respeito da distância  $m$  com relação a uma segunda referência,  $a_2$ , implica em outro círculo que cria uma interseção com o primeiro círculo em precisamente dois pontos. Finalmente, a relação de  $m$  com  $a_3$ , implica em um terceiro círculo em que todos os três círculos interseccionam-se em exatamente um ponto, dando a localização de  $m$ . Esta técnica permite um estudo tri-dimensional, bastando inserir um quarto ponto de referência ao modelo. A conceituação deve iniciar-se com uma esfera associada com o primeiro ponto de referência. Esta esfera deve ser reduzida para um círculo após a introdução de um segundo ponto de referência, reduzindo o problema a aquele de duas dimensões. O modelo é similar ao utilizado no GPS [62].

Este método requer diversos pontos de referência para aumento da precisão e estes pontos devem estar localizados convenientemente, o que não permite que se utilize a infra-estrutura existente de um ambiente sem grandes alterações. Além disso, em casos que se consideram fontes de referência que são próximas umas das outras, o algoritmo acaba por considerar essas informações redundantes.

Outro ponto desfavorável ao algoritmo é o alto custo computacional e a exigência de muitos pontos de referência para o ambiente, sob pena de problemas com singularidade de informações.

Por estes motivos, esse método tradicional não pode ser adicionado aos comparativos com redes neurais artificiais.

### **2.1.2 - Método Probabilístico**

O método probabilístico é baseado em modelos probabilísticos que descrevem a dependência das propriedades do sinal observado na localização do dispositivo móvel. Por meio da modelagem de um mapa de propagação com probabilidade condicional e utilizando o conceito de inferência Bayesiana, o método probabilístico provê uma maneira de lidar com as incertezas e erros inerentes a medidas de sinal [43,60.81]. Discutiremos os princípios básicos desse algoritmo.

Esse método presume um conhecimento prévio da distribuição de probabilidade da localização do usuário. Suponha que conheçamos um conjunto de  $m$  locais onde o usuário possa estar em um dado instante,  $L = \{l_1, l_2, \dots, l_m\}$ . Nesse caso, o modelo estima a distribuição de probabilidade das variáveis de observação ou da variável  $s$  que descreve o sinal por meio da variação de localização  $l$ . Em outras palavras, para qualquer localização  $l$  dada, a distribuição de probabilidade de  $p(s | l)$  que atribui a probabilidade a cada vetor  $s$  do sinal mensurado pode ser obtido. Por meio da aplicação da regra de Bayes, a distribuição posterior da localização é dada por [60]:



$$p(l|s) = \frac{p(s|l)p(l)}{p(s)} = \frac{p(s|l)p(l)}{\sum_{l \in L} p(s|l')} \quad (2.1)$$

$p(l)$  é a probabilidade prévia da localização  $l$  antes de se saber o valor da variável do sinal mensurado, e o somatório é aplicado a todos os valores de possíveis localizações de  $L$ . A distribuição posterior  $p(l|s)$  pode ser utilizada para escolher o estimador ótimo da localização baseada em uma função de perda.

O termo  $p(s|l)$  é conhecido como função de vizinhança. Roos [16] trabalhou com dois métodos de previsão da função de vizinhança: o método de *Kernel* e o Histograma. No primeiro método, o conjunto de probabilidade é designado a um centro (*Kernel*) em torno de cada vetor de amostra no conjunto de treinamento. Deste modo, a densidade resultante estimada para um conjunto de sinais  $s$  na localização  $l$  é uma mistura de uma função de densidade ponderada  $n_l$ , onde  $n_l$  é o número de vetores de treinamento no  $l$ . Suponha que a função de *kernel* seja uma Gaussiana. Sendo assim, a função de vizinhança pode ser escrita como:

$$p(s|l) = \frac{1}{n_l} \sum_{i=1}^{n_l} \left[ \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(s-c)^2}{2\sigma^2}\right) \right] \quad (2.2)$$

onde  $\sigma$  é um parâmetro ajustável que determina a largura do centro. Para extrapolar o método para múltiplas dimensões ou *access points*, Ross multiplica todas as probabilidades condicionais.

Métodos probabilísticos podem fornecer melhor desempenho na estimação da localização se comparado ao algoritmo de triangulação. O método geralmente necessita de um conjunto de treinamento detalhado e extenso para que possa estimar com certa precisão a distribuição de probabilidade condicional.

### 2.1.3 - Algoritmo do Vizinho mais Próximo (*k-Nearest Neighbor*)

#### 2.1.3.1 - Conceitos Básicos sobre Aprendizado de Máquina

A área de aprendizado de máquina tem como objetivos desenvolver técnicas computacionais que permitam simular o processo de aprendizado e a construção de sistemas capazes de adquirir conhecimento automaticamente [49]. Usualmente, algoritmos de aprendizado utilizam experiências anteriores, denominadas casos ou exemplos, para auxiliar o processo de tomada de decisão e melhorar seu desempenho.

De acordo com a característica desses casos ou exemplos têm-se três diferentes modos de aprendizado: supervisionado, não-supervisionado e semi-supervisionado. O que distingue esses modos de aprendizado é a presença ou não do atributo classe, que rotula os exemplos do conjunto de dados fornecido ao algoritmo, denominado conjunto de treinamento. No aprendizado supervisionado, esse rótulo é conhecido, enquanto que no aprendizado não-supervisionado os exemplos não estão previamente rotulados. Já no aprendizado semi-supervisionado, o conjunto de treinamento consiste de uns poucos exemplos rotulados e muitos não rotulados [16].

O conjunto de treinamento para um algoritmo de aprendizado supervisionado consiste, usualmente, de um conjunto  $E$  de  $N$  exemplos (ou casos) de treinamento  $E = \{(x_1, y_1), \dots, (x_N, y_N)\}$  rotulados com os valores  $y$  de uma função  $f$  desconhecida,  $y = f(x)$ , onde os valores  $x_i$  são vetores da forma  $(x_{i1}, x_{i2}, \dots, x_{iM})$  cujos componentes são valores discretos ou contínuos relacionados ao conjunto de atributos  $X = \{A_1, A_2, \dots, A_M\}$ . Ou seja,  $x_{il}$  denota o valor do atributo  $A_l$  do exemplo  $i$ . Dado esse conjunto de exemplos de treinamento, o algoritmo constrói uma hipótese  $hip$  que deve aproximar a verdadeira função  $f$ , tal que, dado um novo exemplo  $\mathbf{x}$ ,  $hip(\mathbf{x})$  prediz o valor  $y$  correspondente. Para valores nominais dos rótulos  $y_1, y_2, \dots, y_N$  o processo é denominado classificação, enquanto que para valores numéricos o processo é denominado regressão [26].

A qualidade de previsão de algoritmos supervisionados é avaliada utilizando um conjunto de exemplos rotulados disjunto do conjunto de treinamento, o qual é denominado de conjunto de teste [26].

#### 2.1.3.2 - Algoritmo $k$ NN – $k$ -Nearest Neighbor

Uma maneira de prever o valor  $y$  de um novo exemplo consiste em comparar esse exemplo com outros cuja classe é conhecida e o atribuir à classe do caso mais próximo. Por exemplo, em um consultório médico, o especialista, a partir do conjunto de sintomas que descrevem o estado de saúde do paciente pode procurar, em fichas médicas de pacientes já diagnosticados, conjuntos de sintomas similares a este, no intuito de auxiliar no diagnóstico de determinada doença [26].

O algoritmo  $k$ -Nearest Neighbor –  $k$ NN – é um algoritmo de aprendizado supervisionado introduzido por D.W. Aha [26,3]. A idéia geral desse algoritmo consiste em encontrar os  $k$  exemplos rotulados mais próximos do exemplo não classificado e, com base no rótulo desses exemplos mais próximos, é tomada a decisão relativa à classe do exemplo não rotulado. Os algoritmos da família  $k$ NN requerem pouco esforço durante a etapa de treinamento. Em contrapartida, o custo computacional para rotular um novo exemplo é relativamente alto, pois, no pior dos casos, esse exemplo deverá ser comparado com todos os exemplos contidos no conjunto de exemplos de treinamento [26].

A figura 2.3 ilustra essa idéia para um problema de classificação, com um conjunto de exemplos de treinamento descrito por dois atributos, no qual, exemplos com rótulo positivo (+) referem-se a pacientes doentes e exemplos com rótulo negativo (-) a não doentes. Considerando o algoritmo KNN para classificação, com  $k=1$ , o novo exemplo  $E_i$  seria classificado de acordo com o único vizinho mais próximo, que é da classe positiva (+) [84].

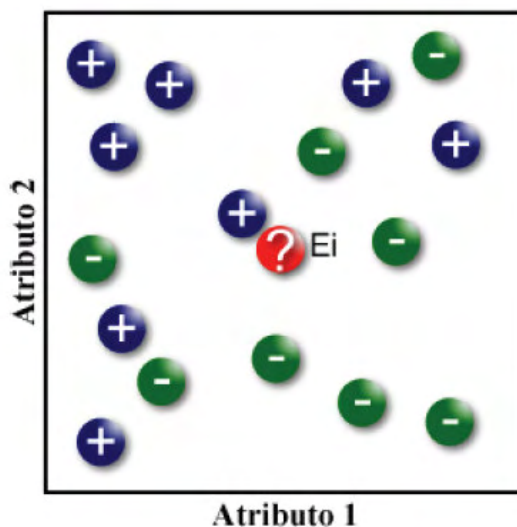


Figura 2.3 – Exemplo de classificação do método *k-Nearest Neighbor* [26]

#### 2.1.4 - Outros métodos clássicos de Localização *Indoor*

Muitos sistemas de localização foram propostos nos últimos tempos, tais como o *Active Badge* [71], que utiliza sinais infravermelhos; *Active Bat* [73,29], *Cricket* [57], que utiliza uma combinação de RF e sinais de ultrassom para estimar distâncias, *PinPoint 3D-iD* [75], *RADAR* [7], *Nibble* [14] e *SpotOn* [32], que utiliza características de sinais de rádio-frequência para estimar localizações. As principais características desses sistemas são discutidas a seguir.

*Active Badge* [71] utiliza uma infra-estrutura de rede que se baseia na tecnologia de infravermelho difusa. Por meio de sensores infravermelhos que são instalados no ambiente, um servidor de localização coleta sinais de infravermelho emitidos por um crachá e utiliza esses sinais para determinar a localização do crachá (que geralmente é utilizado por um funcionário ou visitante). Apesar de a técnica resultar em informações relativamente corretas, o alcance é limitado e o desempenho não é estável por conta da natureza do sinal infravermelho.

O *Active Bat* [73],[29] é um método desenvolvido pelos pesquisadores da *AT&T*. Ele possui uma estrutura bem similar ao *Active Badge*. Um controlador central tem o papel de um servidor de localização. A diferença entre um algoritmo e outro é que o *Active Bat* utiliza tanto rádio como ultrassom ao invés de infravermelho. O controlador envia

um pacote de pedido RF para a tag do *Active Bat* e um pacote de *reset* para os receptores montados no teto ou sensores de ultrassom, ao mesmo tempo. A *tag* responde com um sinal ultrassom. Os receptores calculam a distância baseada na diferença de tempo entre o pacote de *reset* que chega ao receptor e a detecção de sinais de ultrassom. A acurácia e precisão são muito altas, pode alcançar 9 cm em 95% das localizações. Entretanto, o uso de ultrassom para esta aplicação requer uma infra-estrutura de sensor fixo e amplo ao longo de todo o teto. Desse modo, o alcance e a escalabilidade do *Active Bat* são limitados.

O *Cricket* [57], também utiliza ultrassom e sinais RF. Mas o sistema é baseado numa infra-estrutura móvel. Um pequeno dispositivo é anexado ao usuário móvel, o receptor, para estimar a distância para ponto de leitura (*beacon*). O dispositivo móvel é responsável por seu cálculo de localização. O dispositivo móvel mede seu sinal de ultrassom para calcular a distância por meio de tecnologias de *TDOA* (*Time Difference of Arrival - TOA*) e sinais RF para sincronismo, identificação do *beacon* e ativação da medição de tempo. Por conta de o algoritmo utilizar uma infra-estrutura móvel, o *Cricket* beneficia a privacidade, administração descentralizada e baixo custo.

O *3D-iD* [75] é um sistema de localização comercial baseado em RF desenvolvido pela *PinPoint Corp* para determinar a localização 3D de itens dentro de ambientes. O método utiliza uma infra-estrutura de rede. Ele divide o ambiente em células. Cada célula consiste em um controlador central e múltiplas antenas. Estas antenas enviam continuamente (*broadcast*) um sinal. Em consequência da recepção do sinal, uma *tag* anexada ao dispositivo móvel irá, imediatamente, retransmitir a mensagem em outra frequência e codificá-la com seu próprio ID. O controlador da célula mede a distância entre a *tag* e as antenas utilizando o tempo de ida e volta e então, utiliza o método geométrico para determinar a localização da *tag*. O tamanho de célula para boa localização é de 30 m e pode obter de 1m a 3m de acurácia. Entretanto, a célula trabalha na frequência de 2,4 GHz, a *tag* transmite em 5,78 GHz, o que produzirá colisão com o espectro de rádio WLAN. Além do quê, o sistema necessita de muitas células instaladas para cobrir determinado ambiente, e isso definitivamente irá aumentar o custo do sistema por conta do custo do equipamento utilizado.

*RADAR* [7] é desenvolvido pelos pesquisadores da *Microsoft* e é baseado na tecnologia *WLAN*. *RADAR* estima a distância entre *Access points* e dispositivos móveis por meio da medida da intensidade do sinal RF. Um método de propagação baseada em padrões e o algoritmo do vizinho mais próximo são utilizados para calcular a distância do móvel em duas dimensões. Dois métodos são citados em [2] para a criação de um mapa de localização; um é chamado de método empírico, que é baseado na medida de campos, e o outro é baseado no método de propagação baseado em padrões. Os resultados mostram que o método empírico possui um desempenho melhor que o segundo método. Entretanto, o segundo método possui um desenvolvimento mais simples.

*SpotOn* [32] é um sistema de localização *ad hoc*. O método obtém a distância por meio da atenuação da intensidade do sinal ao invés de medidas baseadas em tempo. O *SpotOn* é desenvolvido por meio de dispositivos *RFID* e estações base *AIRID*. É semelhante ao *RADAR* e *3D-iD* no desenvolvimento de uma tecnologia com *tags* baseada em intensidade de sinal RF. Entretanto, experimentos laboratoriais mostram que o *SpotON* pode obter melhor resolução e acurácia que o *RADAR* e com um custo muito mais baixo que o *3D-iD*.

Com isso, concluímos que diferentes tecnologias foram aplicadas na localização *indoor*. Assim como *RADAR*, muitos dos sistemas de localização *indoor* utilizam o método de localização geométrico, por conta *hardware* utilizado, um sistema ultrassom ou com antenas especiais, podem estimar a distância mesmo em locais fechados. Sistemas baseado em ultrassom ou infravermelho podem alcançar uma alta resolução e acurácia, mas possuem desvantagem quanto ao alcance, escalabilidade e custo, tornando a aplicação muito limitada. Sistemas baseados em redes *WLAN*, ao contrário, possuem a vantagem de uma ampla região de cobertura, baixo custo, e uma precisão relativamente boa. Além disso, por conta da complexidade inerente aos ambientes fechados, o desempenho de um sistema de localização vai depender também de como ele é desenvolvido.

## **2.2 - MÉTODOS DE LOCALIZAÇÃO *INDOOR* BASEADOS EM REDES NEURAIIS ARTIFICIAIS**

### **2.2.1 - As Redes Neurais Artificiais**

O desenvolvimento de pesquisas em redes neurais artificiais, geralmente referenciadas como “redes neurais” tem sido motivado desde o seu início pelo reconhecimento de que o cérebro humano processa as informações apresentadas de uma maneira totalmente diferente de um computador digital convencional. O cérebro humano é um computador altamente complexo, não linear e com processamento paralelo (no que tange ao sistema de processamento das informações) [30]. Ele é capaz de realizar certos processos computacionais (como exemplo, o reconhecimento de padrões, percepção, controle motor) com maior eficiência do que as soluções artificiais propostas até aqui.

A idéia principal por trás desta abordagem é a de que computações complexas podem ser obtidas pela combinação de muitos processadores simples altamente interconectados, o que é denominado conexionismo. Substitui-se, nesta abordagem, a arquitetura baseada em um processador central por um sistema de Processamento Paralelo e Distribuído, um dos nomes pelos quais os sistemas conexionistas são conhecidos. Características importantes dos processos cognitivos, como a capacidade de considerar simultaneamente múltiplas restrições ou de combinar múltiplas fontes de conhecimento são representadas com naturalidade nesta arquitetura. Tais características contribuem, por exemplo, para a rapidez e robustez do processo humano de reconhecimento de padrões [58].

### **2.2.2 - Redes Neurais Perceptron Multicamadas (MLP - Multilayered Perceptron)**

Um “perceptron” é uma rede com uma topologia de neurônios dispostos em várias camadas criada por Rosenblatt por volta da década de 1950. A camada de entrada é formada pelos neurônios que recebem diretamente as entradas da rede. Os neurônios que recebem como entradas as saídas daqueles da camada de entrada, constituem a segunda camada e assim sucessivamente até a camada final. Essa camada é conhecida como a camada de saída.

O algoritmo de retropropagação de erro (*error back-propagation*) tem sido aplicado nos perceptrons de múltiplas camadas através de treinamento de forma supervisionada. O princípio que caracteriza o algoritmo é a regra de aprendizagem por correção de erro. Consiste, basicamente, de dois passos através das diferentes camadas da rede, a Propagação (um passo para frente) e a Retropropagação (um passo para trás). Na propagação, um vetor de entrada (padrão de atividade) é aplicado aos nós sensoriais da rede e seu efeito se propaga através da rede, camada por camada.

Finalmente, um conjunto de saídas é produzido como a resposta real da rede. No decorrer do passo de propagação, os pesos sinápticos da rede são todos fixos. No passo para trás, os pesos sinápticos são todos ajustados de acordo com uma regra de correção de erro. A resposta real da rede é subtraída de uma resposta desejada (alvo) para produzir um sinal de erro. Este sinal de erro é então propagado para trás através da rede, contra a direção das conexões sinápticas – o que explica o nome de "retropropagação de erro" (*error back-propagation*). Os pesos sinápticos são ajustados para fazer com que a resposta real da rede se mova para mais perto da resposta desejada, em um sentido estatístico.

#### 2.2.2.1 - Algoritmo de Retropropagação

Neste algoritmo, no passo para frente, os pesos sinápticos se mantêm inalterados em toda a rede e os sinais funcionais da rede são calculados individualmente, neurônio por neurônio. O sinal funcional que aparece na saída do neurônio  $j$  é calculado como:

$$y_j(n) = \varphi(v_j(n)) \quad (2.3)$$

Onde  $v_j(n)$  é o campo local induzido do neurônio  $j$ , definido por [81]:

$$v_j(n) = \sum_{i=0}^m w_{ji}(n)y_i(m) \quad (2.4)$$



Onde  $n$  é o número total de entradas (excluindo o *bias*) aplicadas ao neurônio  $j$ , e  $w_{ij}(n)$  é o peso sináptico que conecta o neurônio  $i$  ao neurônio  $j$ , e  $y_i(n)$  é o sinal de entrada do neurônio  $j$  ou equivalentemente, o sinal funcional que aparece na saída do neurônio  $i$ . Se o neurônio  $j$  estiver na primeira camada oculta da rede,  $n = n_0$  e o índice  $i$  se refere ao  $i$ -ésimo terminal de entrada da rede, para o qual escrevemos:

$$y_i(n) = x_i(n) \quad (2.5)$$

Onde  $x_i(n)$  é o  $i$ -ésimo elemento do vetor (padrão) de entrada. Se, por outro lado, o neurônio  $j$  estiver na camada de saída da rede,  $n = n_L$  e o índice  $j$  se refere ao  $j$ -ésimo terminal de saída da rede, para o qual escrevemos:

$$y_j(n) = o_j(n) \quad (2.6)$$

Onde  $o_j(n)$  é o  $j$ -ésimo elemento do vetor (padrão) de saída. Esta saída é comparada com a resposta desejada  $d_j(n)$ , obtendo-se o sinal de erro  $e_j(n)$  para o  $j$ -ésimo neurônio de saída. Assim, a fase de propagação da computação começa na primeira camada oculta, com a apresentação do vetor de entrada, e termina na camada de saída calculando o sinal de erro de cada neurônio desta camada. O passo de retropropagação, por outro lado, começa na camada de saída passando-se os sinais de erro para a trás através da rede, camada por camada, e recursivamente calculando-se o gradiente local de cada neurônio. Este processo recursivo permite que os pesos sinápticos sofram modificações de acordo com a regra delta. Para um neurônio localizado na camada de saída, o  $\delta$  é simplesmente igual ao sinal de erro daquele neurônio multiplicado pela primeira derivada da sua não-linearidade. Assim, utilizamos a equação abaixo para calcular as modificações dos pesos de todas as conexões que alimentam a camada de saída [47].

$$\left( \begin{array}{l} \text{Correção de} \\ \text{peso } \Delta w_{ji}(n) \end{array} \right) = \left( \begin{array}{l} \text{Parâmetro da} \\ \text{taxa de aprendizagem} \\ \eta \end{array} \right) * \left( \begin{array}{l} \text{Gradiente} \\ \text{local} \\ \delta_j(n) \end{array} \right) * \left( \begin{array}{l} \text{Sinal de entrada} \\ \text{do neurônio } j \\ y_i(n) \end{array} \right)$$

### 2.2.3 - Mapas Auto-organizáveis de Kohonen

Existem diferentes filosofias que determinam as categorias nas quais as arquiteturas de redes neurais e sinais de processos utilizados para modelos sistemas nervosos são divididos [30]. Redes *feedforward* transformam um conjunto de sinais de entrada em outro conjunto de sinais de saída. A transformação de entrada-saída é determinada por ajustes externos e supervisionados dos parâmetros do sistema. Em redes *feedback*, a informação de entrada define o estado inicial da atividade do sistema, e ao final da transição dos estados, o estado final assintótico é identificado como o resultado da computação. Na terceira categoria, células vizinhas da rede neural competem em suas atividades, significando em interações laterais com o objetivo de desenvolver a capacidade de identificar padrões de sinais de entrada. O aprendizado dessa categoria é conhecido como competitivo, não-supervisionado ou auto-organizável.

Um mapa auto-organizável, na forma proposta por Teuvo Kohonen, é um tipo de rede neural artificial, em que suas células são ajustadas de acordo com padrões de sinais de entrada ou classes de padrões por meio de um processo de aprendizado não-supervisionado. Na versão básica, somente uma célula ou grupo de células em um determinado instante de tempo provê uma resposta de ativação para a entrada atual. A localização das respostas tende a ser ordenada em um sistema de coordenadas para diferentes padrões de entrada que foram criados ao longo da rede. A localização espacial ou coordenadas de uma célula na rede corresponde a um domínio específico dos padrões de entrada. Cada célula ou grupo de células agem como decodificadores independentes de uma mesma entrada. É, portanto, a presença ou a ausência de uma resposta de ativação naquela localização, e não somente a transformação do sinal de entrada-saída ou magnitude da resposta que provê a uma interpretação da informação de entrada.

O mapa auto-organizável foi projetado como uma alternativa viável as arquiteturas de redes neurais mais tradicionais. É possível saber o quão “neural” um mapa é. A descrição analítica dessa rede foi desenvolvida tanto na direção técnica quanto biológica. Existem estudos que afirmam que os resultados do aprendizado encontrados aparentam ser tão naturais, que indicam ao menos que o processo de adaptação é semelhante ao cérebro humano.

Mapas auto-organizáveis consistem em mapas, utilizados em tarefas similares a aquelas que redes neurais tradicionais têm sido utilizadas: reconhecimento de padrões, robótica, controle de processos e até mesmo no processamento semântico de informações. A segmentação espacial de diferentes respostas e sua organização em subgrupos topograficamente semelhantes, resultam em um alto grau de eficiência da rede neural no desempenho de operações típicas. [39]

#### 2.2.3.1 - Características básicas do modelo

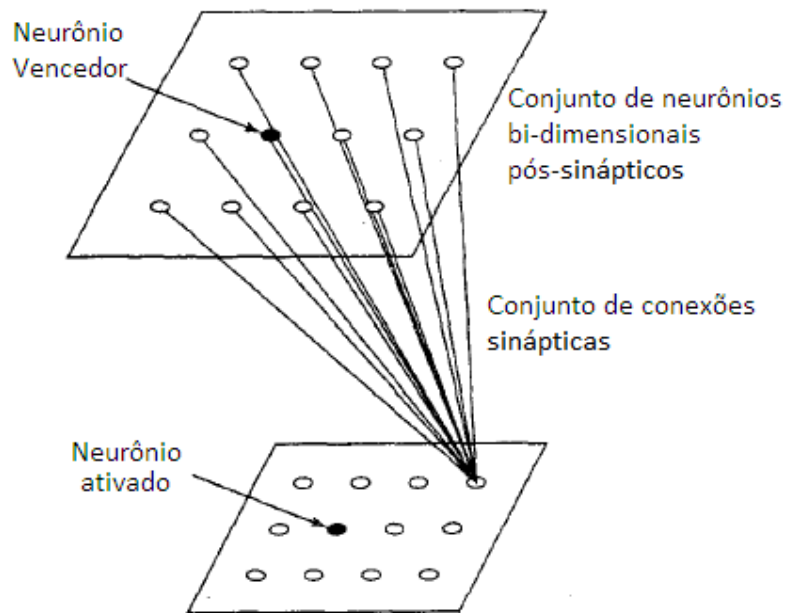
O uso de mapas computacionalmente processados oferece as seguintes características:

- A cada estágio da representação, cada pedaço de informação de entrada é armazenado em seu próprio contexto;
- Neurônios que lidam com pedaços de informação que são relacionadas, são agrupados, assim, podem interagir por meio de conexões sinápticas curtas.

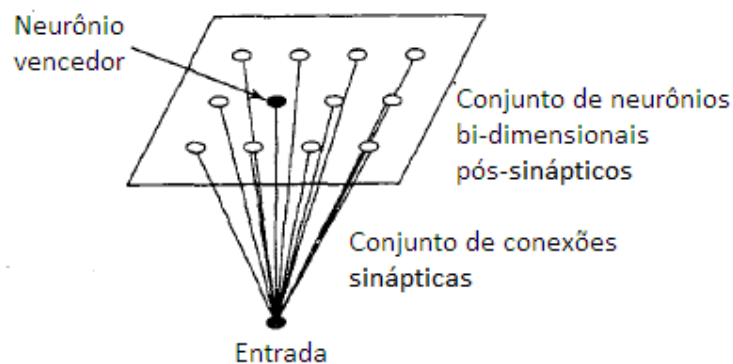
Partindo desse pressuposto, a primeira discussão a respeito de mapas computacionais no cérebro é o princípio da formação topográfica de mapas, discussão essa que foi iniciada por Kohonen:

*“A localização espacial de um neurônio de saída em um mapa topográfico corresponde a um domínio particular ou funcionalidade de dado desenhada no espaço de entrada”. [30]*

Este princípio proveu motivação neurobiológica para dois diferentes modelos de mapeamento descritos a seguir. A figura 2.5 exibe uma configuração de dois modelos. Em ambos os casos, os neurônios de saída são arranjados em um espaço bidimensional. Este tipo de topologia assegura que cada neurônio possua uma série de vizinhos. Os modelos são diferentes um do outro na maneira em que os padrões de entrada são especificados [39].



**(a) Modelo de Willshaw-von der Malsburg**



**(b) Modelo de Kohonen**

Figura 2.4 – Mapas auto-organizáveis [30]

O modelo constante em (a) foi originalmente proposto por Willshaw e Von der Malsburg (1976) em fundamentos biológicos para explicar o problema de mapeamento

da retina para o córtex visual [30]. Uma grade representa neurônios pré-sinápticos e a outra grade representa neurônios pós-sinápticos. As duas grades são interconectadas por sinapses modificadas do modelo Hebbiano. Falando de forma bastante estrita, os neurônios pós-sinápticos não pertencem à categoria “*winner takes all*”, mas, um valor limiar é utilizado para assegurar que somente alguns neurônios pós-sinápticos sejam ativados ao mesmo tempo. A idéia básica de Willshaw-von der Malsburg é baseada na proximidade geométrica dos neurônios pós-sinápticos codificada sob a forma de correlações em relação a sua atividade elétrica, e utilizar estas correlações em grades pós-sinápticas assim como para conectar vizinhos de neurônios pré-sinápticos com os pós-sinápticos.

O segundo modelo (b), proposto por Kohonen, captura as características essenciais de mapas cerebrais e torna-os computacionalmente tratáveis. Aparentemente, o modelo de Kohonen é mais geral que o modelo de Willshaw-von der Malsburg no sentido de que o segundo é capaz de realizar a compressão de dados [30].

#### 2.2.3.2 - Etapas de Funcionamento do algoritmo

O primeiro passo do algoritmo é a inicialização dos pesos sinápticos da rede. Uma maneira bastante utilizada é a atribuição de valores de um gerador de números aleatórios com vistas a evitar que nenhuma série de valores anterior possa ser imposta ao mapa. Após a inicialização da rede, deve-se seguir 3 processos descritos resumidamente a seguir:

1. *Competição*: Os neurônios da rede computam seus respectivos valores por meio de uma função discriminante para cada padrão de entrada. Esta função prove as bases para a competição entre neurônios. Somente um neurônio é declarado o vencedor da competição com base no maior valor da função discriminante.
2. *Cooperação*: O neurônio vencedor determina a localização espacial na topologia da vizinha de neurônios excitados. Esse processo também auxilia na fase de cooperação entre conjuntos de neurônios vizinhos.

3. *Adaptação sináptica*: A adaptação sináptica habilita o aumento dos valores individuais da função discriminante em relação aos padrões de entrada dos neurônios excitados por meio de ajustes aplicados nos respectivos pesos sinápticos.

#### 2.2.3.3 - Iterações

Como já dito anteriormente, uma medida que tem sido adotada por Kohonen é de 500 passos para cada neurônio na camada de saída [66]. Isso significa que, havendo 10 neurônios na camada de saída, deverá haver perto de 5000 iterações para que o mapa esteja ordenado adequadamente. Essas 5000 iterações significam passar 5000 vezes o conjunto de treinamento para a rede neural. Entretanto, ressaltamos que esses parâmetros não são regras. São recomendações resultantes da observação de determinados pesquisadores. Mas, nada impede que se obtenha sucesso com menos ou mais iterações.

#### 2.2.3.4 - O Algoritmo de Treinamento da Rede de Kohonen

Organizando a proposta de uma rede neural de Kohonen sob a forma de um algoritmo computacional, são nove o número total de passos para o treinamento [66]:

01. Inicializar os pesos dos  $N$  neurônios com valores aleatórios baixos (baixos em relação aos valores de entrada).
02. Configurar o raio de vizinhança  $V_i$  de cada neurônio (inicialmente recomenda-se que  $V_i$  seja igual ao tamanho da rede).
03. Apresentar uma entrada à rede.
04. Calcular a distância Euclidiana entre a entrada e os pesos para cada neurônio de saída.

$$d_i(t) = \sum_{j=1}^N (x_j(t) - w_{ij}(t))^2 \quad (2.7)$$

05. Selecionar o neurônio vencedor, ou seja, o neurônio que apresentar a menor distância (o menor  $d_j(t)$ ).

06. Os pesos do neurônio selecionado são atualizados juntamente com todos os neurônios que estão dentro da vizinhança definida por  $V_i(t)$

$$w_i(t+1) = \{w_i(t) + \alpha(t)[x(t) - w_i(t)]\} \quad (2.8)$$

onde  $i \in V_i(t)$  e  $0 \leq \alpha \leq 1$

O termo  $\alpha(t)$  é um termo de avanço que deve ficar entre 0 e 1 ( $0 < \alpha(t) < 1$ ). Recomenda-se iniciar com um valor alto e decrementá-lo com o decorrer do tempo.

07. Se necessário, modificar o raio de vizinhança de todos os neurônios. Essa modificação deverá implicar num decréscimo do raio de  $V_i(t)$ .

08. Se existir, ainda, algum fato que faça parte do conjunto de treinamento que não foi apresentado à rede voltar ao passo 03.

09. Se o número de iterações realizadas até o momento atingiu o número especificado no início, encerrar essa fase do treinamento; caso contrário, realizar nova iteração do conjunto de treinamento retornando ao passo 03.

#### 2.2.3.5 - Atribuição de rótulos

Realizado o ordenamento geral da rede, é preciso saber quais são os neurônios vencedores, para que, apropriadamente, recebam o rótulo devido. Como rótulo,

entendemos o nome do fato [66]. Por exemplo, se o conjunto de treinamento possui três fatos, cada um deles com um nome específico, são certos que os neurônios apontados para cada fato representarão, também, o nome do mesmo. Dessa forma, associamos ao neurônio o nome do fato por ele apontado.

Se os nomes de três fatos treinados fossem "carro", "navio" e "avião", haveria como resultado da rede neural (depois da fase de ordenamento), pelo menos três neurônios, um chamado "carro", outro chamado "navio" e, ainda, outro chamado "avião". Nada impede que mais de um neurônio aponte para o mesmo fato, ou seja, tenha a mesma distância. Acontecendo isto, teremos  $n$  neurônios com o mesmo nome .

Este procedimento é bastante similar ao algoritmo *C-means Clustering* ou agrupamento por C-médias. O algoritmo toma o parâmetro de entrada  $C$ , e divide um conjunto de  $n$  objetos em  $k$  *clusters* (ou partições) tal que a similaridade entraclusters resultante seja alta, mas a similaridade intercluster seja baixa. A similaridade em um *cluster* é medida em respeito ao valor médio dos objetos nestes *clusters* [56].

O *C-means* apresenta bons resultados para conjuntos de dados com *clusters* densos e compactos e bem separados um dos outros, porém, a necessidade de se especificar o parâmetro  $c$  (o número de *clusters*) é visto como uma desvantagem em relação a outros algoritmos de clusterização [56].



### 3 - AS REDES LOCAIS SEM FIO (WLANS)

A tecnologia de localização sem fio se refere basicamente, ao equipamento ou algoritmos que são aplicados em sistemas de comunicações sem fio para determinar sua posição geográfica, e em alguns casos, a velocidade e a direção dos dispositivos móveis. Estudos recentes nessas áreas, em especial baseados na telefonia celular, modems sem fio e serviços e GPS conseguiram que, a técnica de localização de dispositivos móveis em tempo real seja viável e com uma boa relação custo-benefício. Em um futuro próximo, diversos serviços com estes fins surgirão.

Em um sistema de comunicações móveis típico, cada célula de uma torre de transmissão de sinais via rádio se comunica com dispositivos móveis, incluindo Computadores de Mão (*Palmtops*) equipados com conexões sem fio e celulares. Existem dois métodos básicos para a localização de um dispositivo móvel. No primeiro caso, o próprio dispositivo é capaz de determinar sua própria localização por meio de um GPS. Ou, a localização do dispositivo móvel pode ser determinada externamente por meio de seu sinal de rádio, através da medida de intensidade do sinal de chegada tomando como base três ou mais estações base. Quando essas estações calculam a localização do móvel, o método de propagação de sinais conhecido como "*pattern-matching*" é utilizado. Nesse método, o sinal do dispositivo móvel é recebido em diversas células de sites diferentes, e é então transferida para um dispositivo móvel que trabalha como um servidor de localização que tem como objetivo determinar a localização do móvel por meio da medida dos padrões de frequência da transmissão de rádio e as características dos múltiplos caminhos que um sinal vindo de uma simples transmissão pode apresentar ao ser recepcionado em várias e diferentes células em momentos distintos. Esta assinatura de frequência única do sinal é comparada com padrões similares armazenados em uma base central que, conseguirá determinar as coordenadas do dispositivo.

Ao contrário dos sistemas de comunicações móveis, o padrão IEEE 802.11, conhecido como WLAN, geralmente não possuem funções ou métodos para medir, com precisão, o tempo de chegada de um sinal wireless. Sem a ajuda de dispositivo próprios, a única

característica do sinal RF que pode ser utilizada para a estimação da posição é a intensidade do sinal de rádio (*RSS*). Felizmente, o padrão WLAN provê uma forma para que as aplicações de alto-nível possam acessar os parâmetros de intensidade de sinal de baixo nível - o *RSSI* ou indicador da intensidade do sinal. O *RSSI* é um desses parâmetros disponíveis nas interfaces de redes sem fio (*NIC*), e é a única forma que aplicações conseguem obter a informação de intensidade de sinais baseadas nas configurações padrão do IEEE 802.11. Cada ponto de acesso sem fio (*Access Point*) envia sua mensagem de "*beacon*" periodicamente.

Então, uma estação móvel recebe as informações do *RSSI* de todos os *APs* que ela consegue alcançar, independente desse dispositivo móvel ser associado ou não a rede.

Nessa seção, vamos discutir o sistema IEEE 802.11 WLAN e o parâmetro *RSSI* - que é parâmetro presente no padrão IEEE 802.11 para indicar a intensidade de um sinal sem fio.

Como já discutimos anteriormente, o método baseado em redes neurais [9],[10], é um dos métodos mais estudados na atualidade para a estimação da localização de um dispositivo móvel por meio de padrões de propagação de rádio armazenados previamente.

Nessa dissertação, uma solução baseada em redes neurais artificiais será utilizada para extrair a informação de localização por meio do *RSSI* de redes de determinado ambiente.

Na seção seguinte, discutiremos os conceitos inerentes a redes WLANs e suas aplicações no campo da localização em ambientes fechados.

### 3.1 – A FAMÍLIA 802.11

Desde o ano de 1997, o IEEE desenvolve o padrão 802.11 para redes LAN. Os padrões mais significativos incluem o 802.11, 802.11a, 802.11b e 802.11g [36]. IEEE 802.11b opera na faixa de frequência de 2,4 GHz, assim como o 802.11g, enquanto o 802.11a opera na faixa de 5 GHz.

O padrão 802.11 define o controle de acesso de mídia (*MAC - Media Access Control*) e as camadas físicas (*PHY - Physical Layers*) para acesso a redes *ad hoc*. Com isso, o padrão suporta três camadas físicas: infra-vermelho, espalhamento de espectro por salto em frequência (*FHSS - Frequency Hopping Spread Spectrum*) e espalhamento de espectro por sequência direta (*DSSS - Direct Sequence Spread Spectrum*) [23].

A transmissão em RF no padrão 802.11 opera na faixa de 2,4 GHz. A primeira extensão do padrão 802.11, especificação 802.11b, continua operando na faixa de 2,4 GHz com taxas de transmissão da ordem de 1, 2, 5.5 e 11 Mbps. As extensões 802.11a e g são destinadas a transmissões com alta taxa de transferência e utilizam a multiplexação por divisão de frequência ortogonal (*OFDM - Orthogonal Frequency-Division Multiplexing*). O padrão PHY provê taxas de transferências na faixa de 6 a 54 Mbps em 5 GHz e 2,4 GHz respectivamente [23].

#### 3.1.1 - O Indicador de Intensidade de Sinal (*RSSI – Received Signal Strength Indicator*)

O *RSSI* é um dos parâmetros disponíveis em uma interface de redes sem fio. Existem duas subcamadas relacionadas a ele - a subcamada conhecida como Protocolo de Convergência da Camada Física de alto nível (*Higher Physical Layer Convergence Protocol - PLCP*) e a subcamada de Mídia Física Dependente (*Lower Physical Media Dependent - PMD*) no padrão IEEE 802.11. O *RSSI* é a medida na camada PMD da quantidade de energia que a antena utiliza para receber as unidades de dados PLCP corrente (PPDU) [69].

O *RSSI* foi pensado para ser utilizado de forma relativa, uma vez que ele aumenta monotonicamente em função da potência recebida [34].

"O *RSSI* é um parâmetro opcional que possui um valor inicial de 0 até *RSSI\_Max*. Esse parâmetro consiste na medida da subcamada *PHY* da energia que a antena utiliza para receber o *PPDU* atual. A medida do *RSSI* deve ser iniciada entre o início do frame delimitador (*SFD*) e o fim do cabeçalho da verificação de erro (*HEC*). *RSSI* deve ser utilizado de forma relativa. A precisão absoluta do *RSSI* não é especificada." [38]

A figura 3.1 ilustra o formato do quadro *PLCP* no modo *FHSS* (*Frequency Hopping Spread Spectrum*)[34].

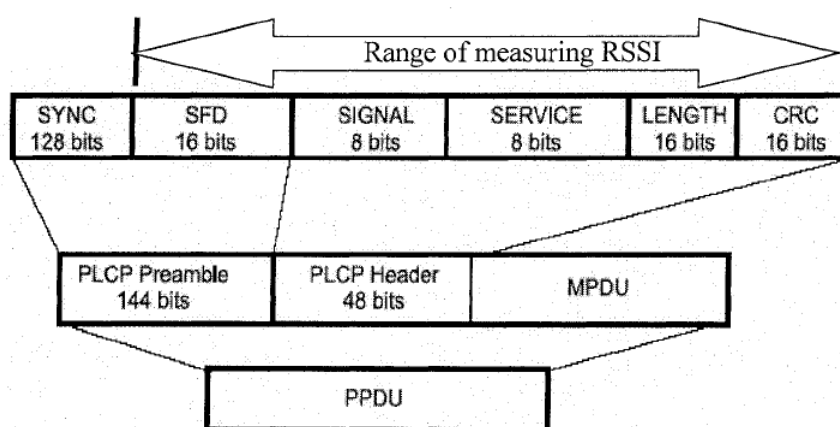


Figura 3.1 – Formato PLCP PPDU [34]

A subcamada *PMD* envia um parâmetro para indicar a quantidade de energia observada na antena selecionada. Esse valor informado é utilizado para gerar o termo *RSSI* na subcamada *PLC* e pode ser utilizado por uma função de diversidade. Se a rede utilizar o *FHSS*, os parâmetros devem estar na faixa de 0 a 15.

Sendo o *RSSI* parâmetro que deve ser utilizado de maneira relativa pela subcamada *MAC*, o parâmetro *PMD* deve ser definido para ter mais do que 16 valores, partindo de 0 até o valor máximo definido pelo usuário (*RSSI\_Max*). O valor 0 é a intensidade de sinal mais baixa possível, enquanto que *RSSI\_Max* é a maior intensidade de sinal. Se a

rede utiliza *DSSS* ou *OFDM*, o valor de *RSSI* deve ser a medida da energia de RF recebida pela camada física do *DSSS* ou *OFDM*.

*RSSI* é representado em 8 bits, e pode suportar até 256 níveis. O valor 0 geralmente é referido como o menor nível de energia RF para o circuito receptor em uma interface 802.11. O menor nível é também chamado de "sensibilidade do receptor", e é uma especificação da interface de rede medida em dBm. Por exemplo, algumas placas de rede sem fio possuem sensibilidade da ordem de -96 dBm. Sendo assim, o valor *RSSI* informado por esta placa será 0 quando a energia de RF medida for menor que -96 dBm.

As medidas do *RSSI* irão variar de 0 a 255 dependendo do fabricante da interface de rede. O valor 1 indica a menor intensidade de sinal detectável pela interface de rede, enquanto que 0 indica nenhum sinal recebido. O valor *RSSI\_Max* é diferente para diferentes fabricantes. Por exemplo, as placas de redes fabricadas pela Cisco Systems retornarão um *RSSI* de 0 a 100. Nesse caso, o *RSSI\_Max* é 100. A placa de rede da Cisco pode informar 101 níveis de potência distintos. Outra placa de rede bastante conhecida é aquela fabricada pela 3Com. Algumas linhas de placas de rede da 3Com retornam valores entre 0 a 60.

A Figura 3.2 ilustra um diagrama do circuito que gera o valor do *RSSI* [33]. A precisão do *RSSI* é definida pelo conversor analógico-digital (ADC). Além de que, a partir do momento em que o *RSSI* é um valor inteiro, ele deve aumentar ou diminuir em passos de números inteiros. Se o *RSSI* muda para 1, significa que o nível de potência foi modificado em alguma proporção na escala de níveis de potências medidas.

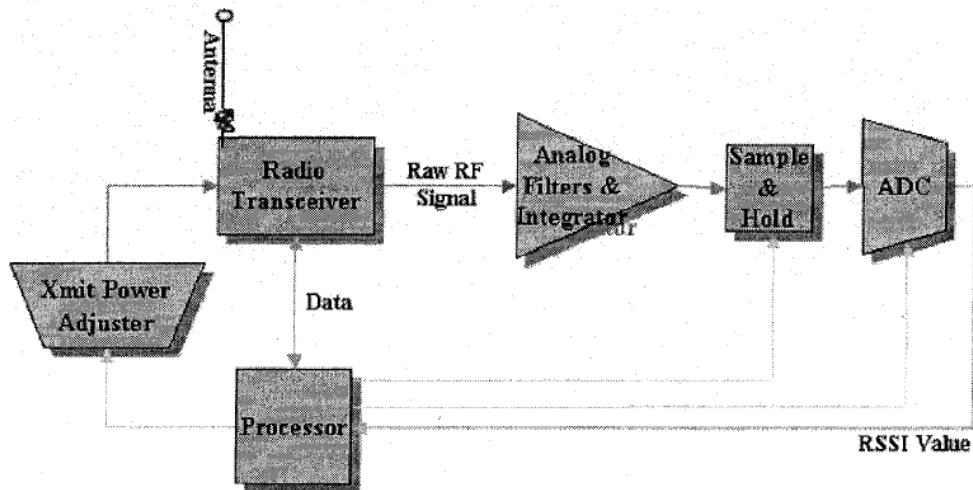


Figura 3.2 - Arquitetura de um circuito para medição do RSSI [33]

### 3.1.2.1 - Fatores que afetam o RSSI

A intensidade do sinal recebido e a faixa de frequência de operação de uma estação de rede sem fio ou *Access Point* são afetadas por diversos fatores [51]. Esses fatores incluem a potência de transmissão (equivalente à potência isotrópica irradiada na antena - EIRP), a sensibilidade de recepção do adaptador *wireless*, o ganho da antena, o padrão de propagação de RF das estações ou *APs* e por fim, possíveis obstruções entre o transmissor e receptor.

A intensidade do sinal recebido é geralmente proporcional à potência transmitida vinda da estação. Tomando como exemplo uma propagação em espaço livre, considere uma transmissão isotrópica de um transmissor como  $P_t$  (medida em Watts).

Em uma distância longa e arbitrária da fonte ( $d$ ), a potência radiada é distribuída uniformemente ao longo de uma superfície de uma esfera. Deste modo, a potência do sinal recebida na distância  $d$  é dada por [51]:

$$P_r = \frac{G_e G_t P_t}{\left(\frac{4\pi d}{\lambda}\right)^2} \quad (3.1)$$

Onde  $G_e$  é o ganho da antena receptora,  $G_t$  é o ganho da antena transmissora,  $\lambda$  é o comprimento da onda eletromagnética.

Fica claro que a intensidade do sinal recebido irá aumentar à medida que a potência de transmissão aumentar. Em se tratando da alocação de frequência e banda, a potência de transmissão é um parâmetro chave que é regulado universalmente. As máximas emissões permitidas no padrão 802.11 variam de região para região [51].

A antena, que é um componente muito importante em um sistema RF, é utilizada tanto para transmitir um sinal como dar forma e foco ao sinal recebido. O ganho da antena representa a capacidade que a antena têm de aumentar a potência efetiva do sinal em uma direção em especial, com o dBi (decibels relativos a um transmissor isotrópico) exercendo a função de unidade de medida. O dBi representa o ganho da antena comparado a um radiador isotrópico, que transmite sinais RF em todas as direções com igual intensidade. Mais precisamente, dBi é igual a 10 vezes o logaritmo (base 10) da intensidade do campo eletromagnético da direção escolhida pela antena, dividido pela intensidade do campo eletromagnético de uma antena isotrópica com medidas tomadas a uma mesma distância [45].

Todas as antenas possuem padrões de radiação que indicam a potência radiada em qualquer direção relativa a direção da máxima radiação. A figura 3.4 ilustra um exemplo de um padrão de radiação de uma antena [45].

Uma antena isotrópica é definida como "uma antena hipotética sem perdas com igual radiação em todas as direções." Percebemos claramente que uma antena desse tipo é uma entidade ideal, onde até mesmo a antena mais simples possui algum nível de diretividade.

Apesar de hipotética e fisicamente não realizável, um radiador isotrópico é referenciado como um meio de expressar as propriedades de diretividade das antenas atuais. Uma antena direcional é aquela em que "possui propriedades de radiação e recepção de ondas eletromagnéticas de forma mais efetiva em algumas direções do que em outras." O

termo é aplicado em antenas onde a máxima diretividade é significativamente maior que uma antena de dipolo linear [51].

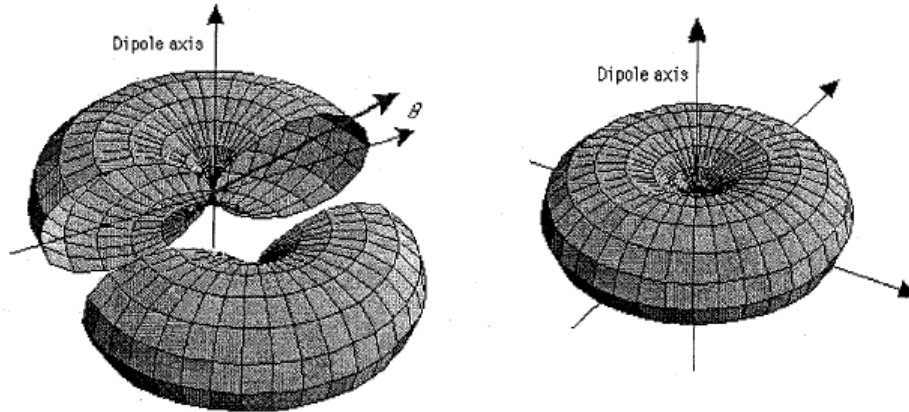


Figura 3.3 - Exemplo do padrão de radiação de uma antena omni-direcional [51]

Um dipolo linear é um exemplo de uma antena omni-direcional - uma antena que possui um padrão de radiação que não é direcional em um plano. A figura 3.3 indica que um dipolo linear possui uma potência uniforme dissipada em qualquer plano perpendicular ao eixo do dipolo, e a maior potência dissipada está localizada no plano equatorial.

Geralmente, a antena direcional pode transmitir o maior nível de potência de um sinal assim como possui a capacidade de receber o menor nível de potência de sinal. Contudo, antenas omni-direcionais são relativamente acessíveis, e a maioria dos adaptadores sem fio possui este tipo de antena montadas nesses equipamentos.

Além da antena, o item mais crítico em sistemas de comunicações sem fio é o receptor utilizado. De maneira especial, é importante deixar claro a importância da sensibilidade do receptor. Essa sensibilidade é o menor nível que um sinal pode ser decodificado por um receptor. A menor sensibilidade recebida está relacionada com o quão extensa é a faixa de cobertura da rede. Quando um sinal é recebido na subcamada *PMD* de uma placa de rede sem fio, o *RSSI* do sinal recebido deve ser comparado com um valor de limiar (*threshold value*), que é especificado no padrão 802.11 como *ED\_THRESHOLD*. Se o *RSSI* é menor que esse valor de limiar, o sinal recebido é descartado. Esse limiar é quem define a sensibilidade do receptor.



Como já discutido anteriormente, a intensidade do sinal recebido no destino é fortemente afetada pelo comportamento do sinal no ambiente onde receptor e transmissor estão localizados. O meio de comunicação de radiofrequência (RF) para residências, escritórios e indústrias são bem diferentes, nem ao menos são parecidos.

Multipercursos e perdas de propagação são questões que devem ser consideradas no projeto de sistemas baseados no padrão IEEE 802.11. A superfície dos móveis, materiais dos elevadores, máquinas e construções de metal que podem obstruir o caminho contribuem para o espalhamento causado pelo fenômeno de multipercurso.

A perda de propagação ocorre quando a potência do sinal é perdida a medida que a distância entre transmissor e receptor aumenta. Isto ocorre por conta da combinação da atenuação causada por paredes, tetos e mobília. Cada parede construída com concreto pode atenuar até 6 dB, e paredes construídas com tijolos, 4 dB.

O desvanecimento provocado pelo multipercurso é outro fator chave que determina as perdas de propagação. O desvanecimento provocado pelo multipercurso ocorre quando o sinal refletido de janelas, paredes, mobília e pessoas provocam um espalhamento do sinal transmitido.

#### 3.1.2.2 - Aplicações do *RSSI*

No padrão 802.11, o *RSSI* é um inteiro arbitrário, que foi originalmente criado para uso interno do microprocessador do adaptador e do *driver* do dispositivo. O *RSSI* pode ser utilizado no adaptador para determinar o instante em que a quantidade de energia do rádio no canal encontra-se abaixo de determinado limiar, onde nesse instante, a interface de rede está pronta para iniciar a transmissão de dados (*CTS – Clear to Send*).

A especificação da camada de controle de acesso para o padrão 802.11 possui diversas semelhanças com o padrão 802.3 (*Ethernet Wired Line*). O protocolo 802.11 é um padrão que possui a funcionalidade de detecção de portadora para múltiplos acessos e prevenção de colisões (*CSMA/CA - Carrier-Sense Multiple Access, Collision Avoidance*).

Este protocolo evita a colisão antes da transmissão dos sinais ao invés de detectar a colisão por meio da monitoração dos canais. A camada *MAC* opera juntamente com a camada física por meio da amostragem da energia transmitida a medida que os dados são transmitidos.

A camada física utiliza um algoritmo conhecido como *clear channel assessment* para determinar quando o canal está pronto para transmissão. Este momento é identificado por meio da medida da energia de RF ou *RSSI* na antena e assim, determina-se a intensidade do sinal recebido. Se o *RSSI* está abaixo de um limiar pré-definido, o "*Clear Channel Threshold*", o canal é declarado pronto para transmissão e a camada *MAC* recebe a confirmação para transmissão.

O limiar conhecido como "*Roaming Threshold*" no padrão 802.11 também utiliza as informações do *RSSI* para determinar o momento em que o cliente deve proceder com a mudança (*handover*) para outro *AP*.

Como o *RSSI* é um parâmetro que ajuda a definir a qualidade do canal entre origem e destino, e também é relacionado com as propriedades de distância e do canal, o *RSSI* é muito útil no gerenciamento de localização de dispositivos móveis, roteamento e modelagem da distribuição de energia de rádio-frequência [19,50]. Como exemplo, o método *SpotON* [32] é um sistema utilizado para analisar a sensibilidade na localização na configuração *ad hoc*.

As marcas do *SpotON* utilizam a informação de intensidade do sinal recebido como um estimador de distância. As localizações obtidas podem não ser exatamente a localização física dos nós ou estações, mas a distância descreve a qualidade da conexão [32].

A informação da localização é útil para roteamento. Pode ser utilizada como suporte a criação da tabela de roteamento dessas redes.

O *RSSI* é útil para estimar as propriedades de um canal e na modelagem de um canal em específico. Juntamente com outros parâmetros de qualidade de sinal, o *RSSI* pode

também prover meios para a atribuição de canais de transmissão [44] ou até mesmo para o gerenciamento inteligente do canal.

## 4 - METODOLOGIA EMPREGADA

### 4.1 - ESTIMAÇÃO DO MODELO

#### 4.1.1 - Aspectos computacionais

Este trabalho utilizou software especializado de simulação de matrizes para todas as implementações apresentadas. No que tange aos aspectos computacionais disponíveis para essas simulações, a tabela 4.1 resume os principais deles:

Tabela 4.1 - Quadro resumo contendo os aspectos computacionais empregados no trabalho

Aspectos computacionais – Quadro Resumo	
<b>Estação para testes</b>	Notebook Sony Vaio VGN-CR260A
Processador	Intel Core 2 Duo T7250 (2 GHz)
Cache	2048 MiB
Memória	2GiB DDR2 (400 MHz Dual Channel)

#### 4.1.2 - Critérios para escolha e descrições do ambiente destinado aos testes experimentais

Primeiramente, o local deve ser em grande parte fechado, assim problemas como dispersão do sinal elétrico e interferências associadas são minimizadas. Outra característica importante é a disponibilidade de vários *APs* de conexão sem fio (*Access Points - APs*) com características diferentes. Estas exigências, simulam uma situação real ao qual o modelo proposto poderá ser empregado, no qual servem como base para a mensuração da convergência do algoritmo proposto e do total de erros na etapa de simulação (eficiência da rede). Por fim, o ambiente deve pertencer à estrutura da UnB e ser de fácil acesso para medidas experimentais.

De posse destes pré-requisitos, o ambiente selecionado é o Prédio da Divisão Técnica Laboratorial da Engenharia Elétrica – SG11. O prédio é baseado no projeto de João Figueiras Lima, foi construído entre os anos de 1964 e 1965 com área total edificada de  $3.516,2200m^2$ . O edifício é modulado, de dois andares e subsolo,

construído em estrutura e placas pré-moldadas. Internamente, os ambientes são divididos por placas removíveis. A sobreloja desmontável se apóia na cobertura por meio de tirantes. Este projeto foi repetido em três construções, os SGs 9, 11 e 12. Os SGs 9 e 12 foram construídos realmente de acordo com o projeto, que previa a utilização de pré-moldados, mas o SG 11 foi construído em alvenaria convencional, pois na época, só havia uma empresa capaz de fornecer as peças, o que caracterizaria vício de licitação. [24]

O **primeiro piso** contempla os seguintes laboratórios:

- Laboratório de Pesquisa de Qualidade de Energia (Área = 39,27  $m^2$ );
- Laboratório de Ensino de Máquinas (Área = 83,05  $m^2$ );
- Laboratório de Pesquisa de Fontes Alternativas de Energia (Área = 34,48  $m^2$ );
- Laboratório de Ensino de Conversão de Energia Elétrica (Área = 93,98  $m^2$ );
- Laboratório de Ensino de Instalações Elétricas e Eletricidade (Área = 95,39  $m^2$ );
- Laboratório de Pesquisa de Tratamento de Superfícies e Dispositivos (Área = 54,92  $m^2$ );
- Laboratório de Projeto de Circuito Integrado (Área = 22,33  $m^2$ );
- Grupo de Processamento Digital de Sinais (Área = 61,71  $m^2$ );
- Laboratório de Automação Bancária;
- Laboratório de Ensino de Controle Servomecanismo;
- Laboratório de Análise Dinâmica Linear.

Total da área para o primeiro piso: 365,20  $m^2$ . Todos os laboratórios citados anteriormente foram objeto de estudo deste trabalho.

O **segundo piso** contempla os seguintes ambientes:

- Gerência e Sala de Reuniões (Área = 38,41  $m^2$ );

- Sala de Funcionário (Área = 26,85  $m^2$ );
- Copa (Área = 7,83  $m^2$ );
- Laboratório de Simulação de Sistemas (Área = 37,36  $m^2$ );
- Laboratório de Sistemas Digitais (Área = 52,69  $m^2$ );
- Laboratório de Antenas e Eletromagnetismo (Área = 75,63  $m^2$ );
- Laboratório de Telecomunicações (Área = 64,30  $m^2$ );
- Laboratório de Automação, Visão e Sistemas Inteligentes;
- Laboratório de Pesquisa, Robótica e Automação;
- Laboratório de Pesquisa, de Controle e Visão por Computador;
- Laboratório de Sistemas Digitais, Arquitetura de Processadores;
- Laboratório de Circuitos Elétricos I e II Aplicados, Telecomunicações e Eletrônica.

Total da área para o segundo piso: 365,20  $m^2$ . Todos os laboratórios/ambientes citados anteriormente foram objeto de estudo deste trabalho.

#### **4.1.3 - Critérios para escolha dos *Access Points* – APs**

O SG-11 dispõe de mais de 14 APs sem fio de diversos fabricantes e com características distintas cada um. É importante que sejam escolhidos APs cujo sinal seja alcançado na maior parte do prédio em ambos os andares. Partindo desse pensamento, observou-se que, apesar da grande quantidade de opções de escolha, somente quatro emissores atenderam a essa premissa.

Entendemos que apesar de somente quatro desses emissores terem seus sinais alcançados em toda parte da planta do prédio, podíamos utilizar os demais APs em algumas regiões da planta. Entretanto, propositalmente esse não foi o caminho adotado. O objetivo é de se evitar que as simulações tivessem pontos/regiões com inconsistências de dados, uma vez que esses sinais não seriam uma boa referência para grande parte da planta. Além disso, busca-se criar uma rede neural uniforme em toda a sua extensão da planta, o que não seria possível com a adoção em massa de todos os *Access Points* disponíveis no ambiente.

Com isso, temos os seguintes *APs* de referência com sua respectiva localização na planta (Tabela 4.2):

Tabela 4.2 - Quadro resumo contendo a identificação e localização dos *APs* utilizados no experimento

<i>APs</i> utilizados no experimento	
<b>Nome</b>	<b>Localização na Planta</b>
WLPCI	Térreo
LabFontes	Térreo
GRAV	Sobreloja
LAVSI	Sobreloja

A configuração encontrada, além de atender aos requisitos do problema, é sobremaneira interessante, uma vez que uma parcela igual de emissores encontra-se em cada andar. Isto é, **WLPCI** e **LabFontes** encontram-se no térreo do prédio e em posições distintas e **GRAV** e **LAVSI**, na sobreloja do SG11 (Figuras 4.1 e 4.2).

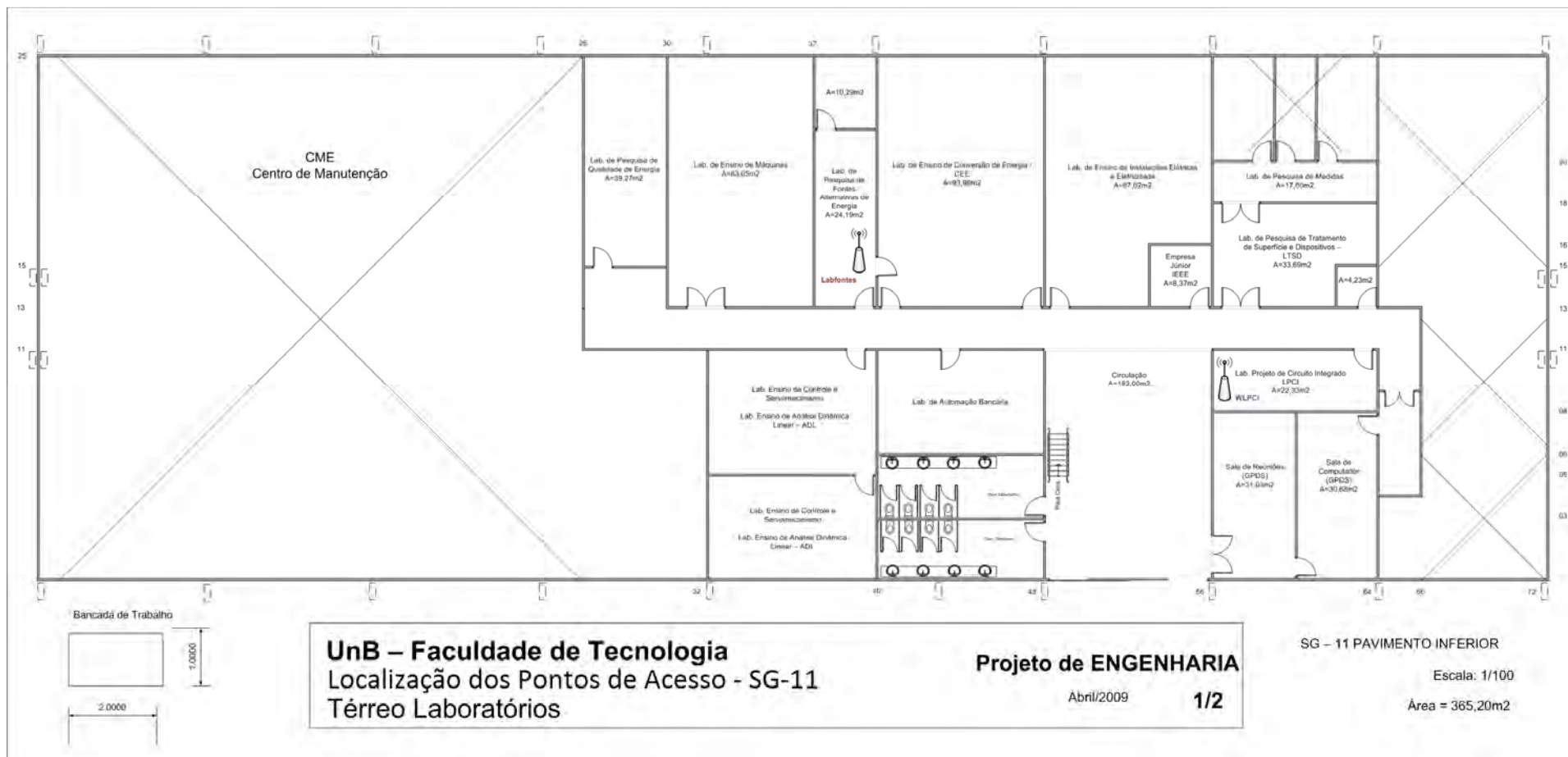


Figura 4.1 - Localização dos APs SG-11 – Térreo– Escala 1/100



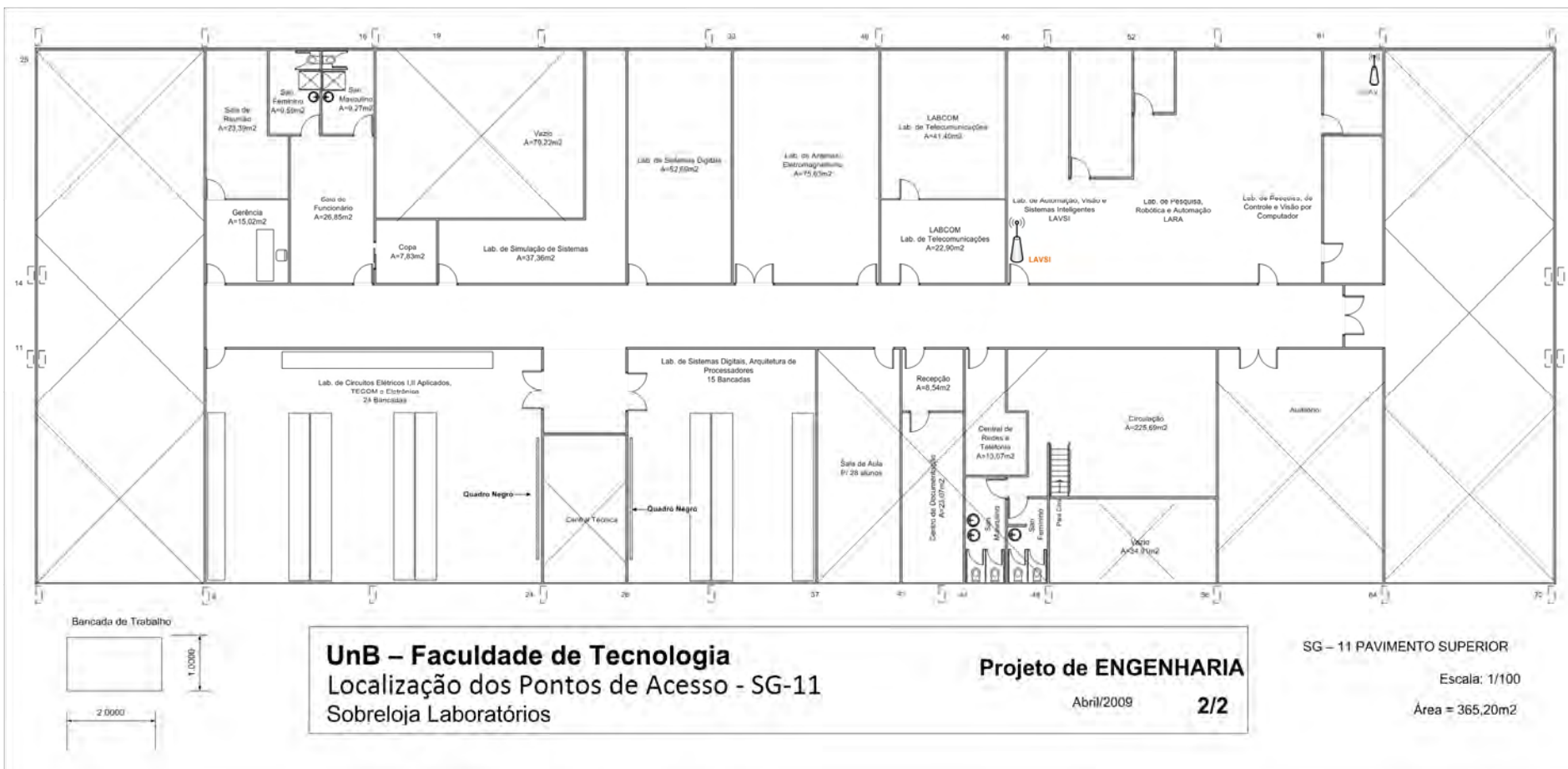


Figura 4.2 - Localização dos APs SG-11 – Sobreloja – Escala 1/100

#### 4.1.4 - Aquisição experimental dos valores do sinal recebido (RSSI)

O trabalho em campo consistiu na verificação do valor do *Received Signal Strength Indicator* – RSSI para os pontos dentro de cada área de interesse através de software especializado.

Por conta da alta sensibilidade, grande disponibilidade de opções, software baseado em código aberto (dispensando-se a necessidade de aquisição de licenças de uso) e aceitação no mercado (Vencedor do Prêmio *InfoWorld Best of Open Source Software Awards - 2008*), o aplicativo **inSSIDer** da fabricante **MetaGeek** – disponível em: [www.metageek.net/products/inssider](http://www.metageek.net/products/inssider) - foi utilizado para as medições adquiridas pelo notebook. Para validação das medidas realizadas com o notebook, foi utilizado o aplicativo **Barbelo** – disponível em [www.darkircop.org/barbelo/](http://www.darkircop.org/barbelo/) - em um dispositivo celular da marca Nokia modelo N95 III geração.

O processo operacional de medidas baseou-se em cinco verificações do valor RSSI de cada ponto em cada AP com um intervalo de 10 segundos entre cada uma das 5 medidas. Sendo assim, para 01 ponto, temos 5 medidas para cada *access point* x 4 APs de referência, totalizando 20 medidas.

Ao final do processo, um mapa de propagação foi constituído com as seguintes características (Tabelas 4.3 e 4.4):

Tabela 4.3 - Quadro resumo das medidas experimentais

Quadro resumo das medidas experimentais	
Total de Pontos (Sobreloja)	118
Total de Pontos (Térreo)	98
<b>Total de Pontos</b>	<b>216</b>
Quantidade de Medidas (Sobreloja)	2360
Quantidade de Medidas (Térreo)	1960
<b>Quantidade de Medidas</b>	<b>4320</b>

Tabela 4.4 - Distribuição dos Pontos entre Sobreloja e Térreo – SG 11

Distribuição dos Pontos		
	Pontos	Percentual
Pontos na Sobreloja	118	54,62 %
Pontos no Térreo	98	45,38 %
<i>Totais</i>	<i>216</i>	<i>100,00 %</i>

Em seguida, estas medidas foram armazenadas, juntamente com a coordenada de cada ponto, em um banco de dados, onde posteriormente, foram utilizadas como massa de dados para treinamento, validação e parte da simulação das redes neurais propostas.

As figuras 4.3 e 4.4 ilustram parte do processo de aquisição experimental do RSSI.

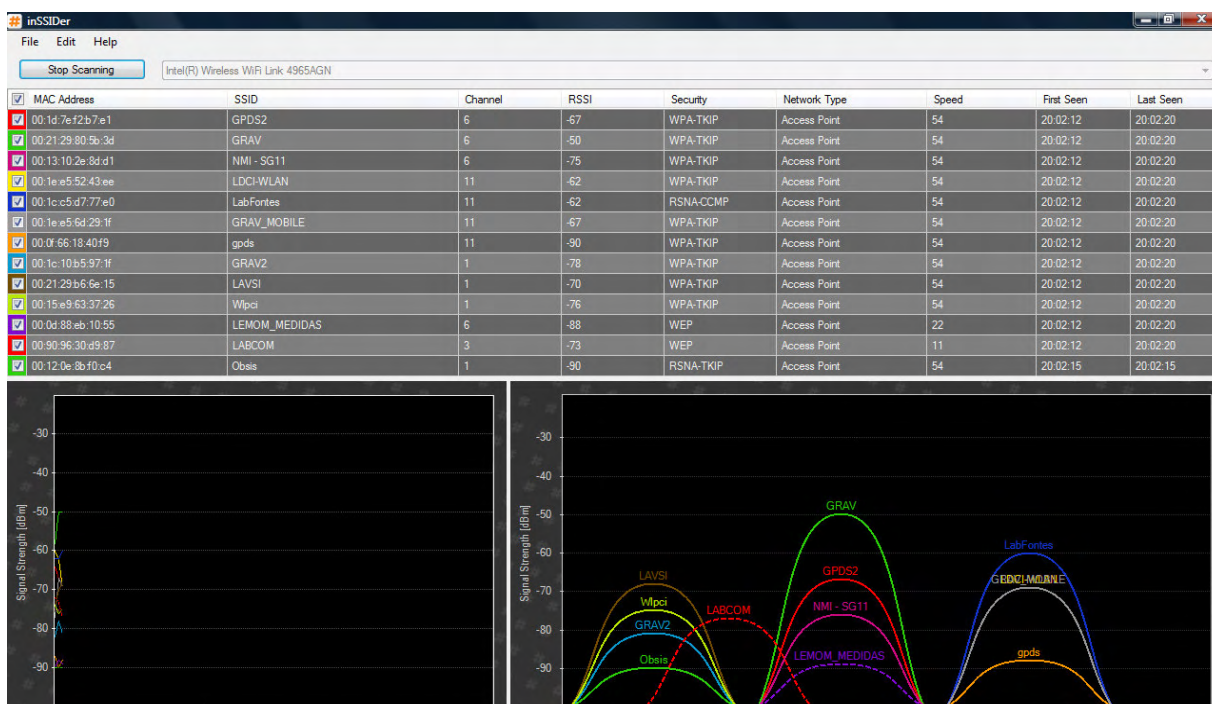


Figura 4.3 - Software de medição inSSIDer contendo todos os APs encontrados no SG-

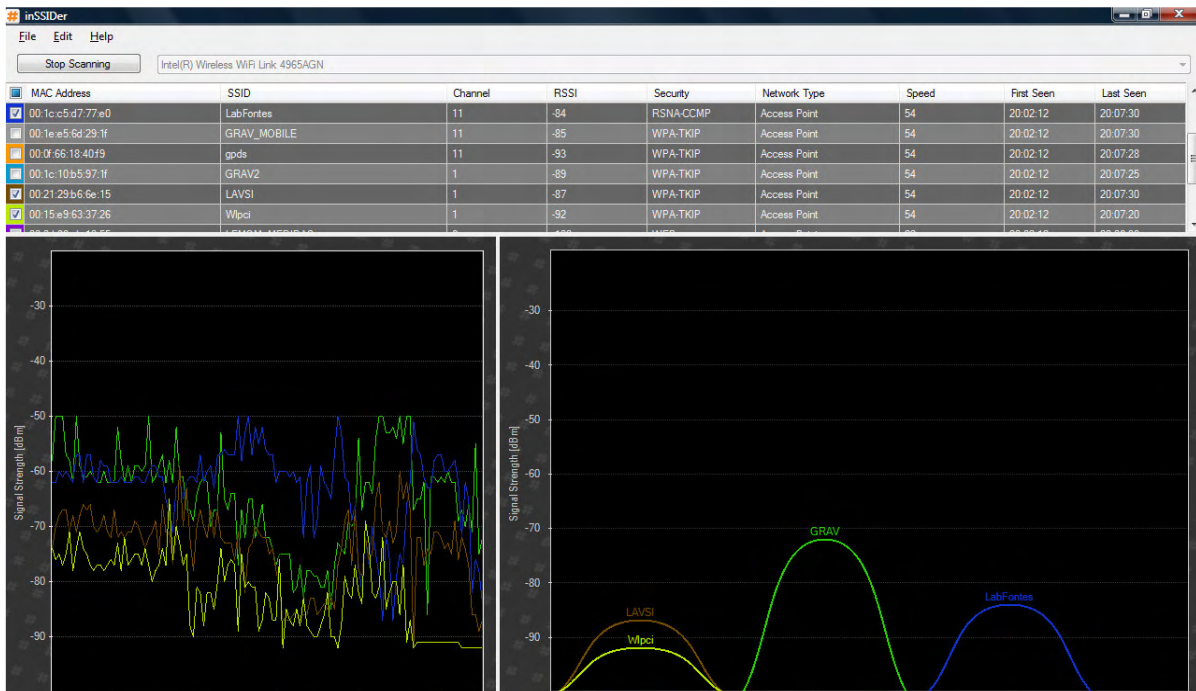


Figura 4.4 - Etapas da medição utilizando o inSSIDer para os APs de referência. Todas as medidas foram realizadas com as portas fechadas e sem pessoas transitando ao longo dos corredores/laboratórios.

#### 4.1.5 - Análises das medidas experimentais e pré-processamento dos dados

Conforme exposto anteriormente, o processo de medidas experimentais gerou um número total de 4.320 amostras divididas em 2 ambientes (SG-11 Sobreloja e Térreo).

O valor médio da intensidade do sinal, desvio padrão, maior e menor valores para cada Ponto de Acesso ficaram assim divididos (antes do tratamento dos dados experimentais) conforme a tabela 4.5, a seguir:

Tabela 4.5 - Quadro resumo das medidas experimentais expressas em dBm

	Medidas expressas em dBm			
	WLPCI	LabFontes	GRAV	LAVSI
Valor Médio	-77,57	-77,18	-73,75	-77,60
Desvio Padrão	-10,52	-11,58	-12,53	-13,47
Menor	-88,09	-88,76	-86,27	-91,07
Maior	-67,05	-65,60	-61,22	-64,13

Este trabalho é baseado em simulações e medidas experimentais para observação de como a intensidade do sinal de quatro APs podem auxiliar na localização de um dispositivo móvel. Para tanto, o primeiro passo neste processo é a criação de um mapa de propagação com as medidas originais. Os resultados devem armazenados de forma organizada, interpretados e finalmente, as medidas que apresentaram erros em sua aquisição experimental foram removidas da massa de dados. Com isso, o mapa será refinado, retirando erros de medidas e auxiliando na análise dos dados coletados.

Optamos pela organização dos dados da forma ilustrada pela tabela 4.6:

Tabela 4.6 - Análise dos dados obtidos para o ambiente “Gerência”

		Medidas expressas em dBm			
		AP1	AP2	AP3	AP4
Local	Dados	WLPCI	Labfontes	GRAV	LAVSI
Gerência	Valor Médio	-85,62	-95,82	-83,89	-98,73
	Desvio Padrão	0,89	4,98	0,31	1,33
	Menor	-86,00	-99,00	-84,00	-99,00
	Maior	-83,00	-87,00	-83,00	-92,00

Onde:

- **Valor Médio** é a média dos valores de determinado AP para o ambiente (No exemplo é a média dos valores dos AP1, AP2, AP3, AP4 para o ambiente Gerência);

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i = \frac{1}{n} (x_1 + \dots + x_n) \quad (4.1)$$

$\bar{x}$  - Valor Médio

$x_i$  - Valor de cada evento individual

$n$  - Total de Elementos

- **Desvio Padrão** é o desvio padrão dos valores de determinado AP para o ambiente;

$$\sigma = \sqrt{\frac{\sum (X_i - \bar{X})^2}{n-1}} \quad (4.2)$$

$\sigma$  - Desvio Padrão

$X_i$  - Valor de cada evento individual ( $X_1, X_2, X_3, X_4, \dots, X_n$ )

$\bar{X}$  - Média aritmética dos valores  $X_i$

$n$  - Total de elementos

- **Maior** é o maior valor encontrado antes do tratamento dos dados do ambiente;
- **Menor** é o menor valor encontrado antes do tratamento dos dados do ambiente.

De posse da tabela 4.6, realizamos um tratamento dos dados buscando tratar os erros de medida experimental para todas as medidas realizadas nas plantas. Com isso, para o ambiente “Gerência” (exemplo analisado), temos o seguinte resultado (Tabela 4.7):

Tabela 4.7 - Dados após a aproximação, para o ambiente “Gerência”

		Medidas expressas em dBm			
		AP1	AP2	AP3	AP4
Local	Dados	WLPCI	Labfontes	GRAV	LAVSI
Gerência	Valor Médio	-85,80	-96,69	-84,09	-98,95
	Desvio Padrão	0,40	3,55	0,29	0,23
	Menor	-86,00	-99,00	-85,00	-99,00
	Maior	-85,00	-91,00	-84,00	-98,00

Tabela 4.8 - Quadro resumo das medidas experimentais expressas em dBm após filtragem dos erros de medida

	Medidas expressas em dBm			
	WLPCI	LabFontes	GRAV	LAVSI
<i>Valor Médio</i>	-78,04	-77,66	-73,23	-78,62
<i>Desvio Padrão</i>	11,32	10,94	12,79	11,67
<i>Menor</i>	-89,36	-88,61	-86,01	-90,28
<i>Maior</i>	-66,72	-66,72	-60,44	-66,95

Por fim, os dados tiveram seus valores normalizados baseados em uma distribuição caracterizada pela média e desvio padrão. A normalização tem como objetivo auxiliar na velocidade de convergência da rede neural e precisão dos resultados apresentados.

$$Z = \frac{X - \bar{x}}{\sigma} \quad (4.3)$$

$Z$  – Valor normalizado

$X$  – Valor que se deseja normalizar

$\bar{x}$  - Média

$\sigma$  - Desvio padrão

## **5 - RESULTADOS OBTIDOS**

### **5.1 - REDE NEURAL DE KOHONEN**

#### **5.1.1 - Treinamento**

Uma propriedade que é de grande importância para uma rede neural é a capacidade que ela possui de aprender a partir do exemplo. Como já discutido nas seções anteriores, uma rede neural aprende por meio de um processo interativo baseado em ajustes de pesos sinápticos e níveis de viés (*bias*).

Este processo só é possível graças ao treinamento, que é basicamente a apresentação dos dados de entrada à rede até atingir um critério de parada, seja ele o número de iterações de treinamento ou o alcance de determinado percentual de erro.

O problema em questão exigiu 03 abordagens distintas:

1. Treinamento contendo as medidas para a Planta Superior do SG-11;
2. Treinamento contendo as medidas para a Planta Inferior do SG-11;
3. Treinamento contendo as medidas para a Planta Completa (Superior+Inferior) do SG-11.

Essa abordagem foi necessária como forma de se observar o comportamento da rede em cada parte da planta, o que apontaria para a necessidade de novas medidas experimentais ou ajustes de parâmetros no treino para a correta convergência do algoritmo de treinamento.

##### **5.1.1.1 - Definição dos Parâmetros para Treino**

O algoritmo de Kohonen exige diversos parâmetros para a correta inicialização e treino com os valores de entrada. Os seguintes parâmetros foram adotados para os treinamentos de acordo com a abordagem:



### **Abordagem 01 (Treinamento com a Planta Superior – SG11)**

Parâmetros:

- **Dimensão do Mapa Auto-organizável:** 2 x 5
- **Taxa de Aprendizado (Fase de Ordenamento):** 0,9
- **Quantidade de Passos de Ordenamento:** 1.000
- **Taxa de Aprendizado:** 0,01
- **Distância da Vizinhaça (Baseada no tamanho do mapa):** 3,0
- **Número Máximo de Iterações:** Treinamentos com 100, 250, 500, 1.000, 2.500, 5.000 iterações
- **Percentual de Erro desejado:** 0 %

### **Abordagem 02 (Treinamento com a Planta Inferior – SG11)**

Parâmetros:

- **Dimensão do Mapa Auto-organizável:** 2 x 5
- **Taxa de Aprendizado (Fase de Ordenamento):** 0,9
- **Quantidade de Passos de Ordenamento:** 1.000
- **Taxa de Aprendizado:** 0,01
- **Distância da Vizinhaça (Baseada no tamanho do mapa):** 3,0
- **Número Máximo de Iterações:** Treinamentos com 100, 250, 500, 1.000, 2.500, 5.000 iterações
- **Percentual de Erro desejado:** 0 %

### **Abordagem 03 (Treinamento com a Planta Completa – SG11)**

Parâmetros:

- **Dimensão do Mapa Auto-organizável:** 4 x 5
- **Taxa de Aprendizado (Fase de Ordenamento):** 0,9
- **Quantidade de Passos de Ordenamento:** 1.000
- **Taxa de Aprendizado:** 0,01
- **Distância da Vizinhaça (Baseada no tamanho do mapa):** 3,0
- **Número Máximo de Iterações:** Treinamentos com 100, 250, 500, 1.000, 2.500, 5.000, 7.500, 10.000 iterações
- **Percentual de Erro desejado:** 0 %

É importante frisar que este estudo seguiu as recomendações presentes na literatura técnica quanto à **taxa de aprendizado** (0,9 para ordenamento e 0,01 na fase de convergência), **passos de ordenamento** (valor típico de 1.000), **distância da vizinhança** (de acordo com a dimensão do mapa) e **número máximo de iterações** (500 vezes o número de neurônios da rede) [66].

#### 5.1.1.2 - Análise preliminar (pré-treino) dos dados de entrada e possíveis agrupamentos

Antes de darmos início ao processo de treinamento, realizamos uma análise preliminar dos dados de entrada para as duas plantas com o objetivo de prever possíveis agrupamentos de amostras com características semelhantes.

Os resultados desses agrupamentos nos deixaram surpresos em especial as medidas de *APs* localizados em andares diferentes. Isto é, quando analisávamos dados de *APs* localizados no andar térreo, estando na sobreloja do SG-11, como exemplo. O que ocorre é que, estando estes *APs* localizados em andares diferentes, os fenômenos típicos que contribuem para a degradação de um sinal eletromagnético agiram de forma bastante intensa no intervalo de intensidade de sinal de cada *AP* e principalmente no agrupamento de ambientes em categorias próximas ou até mesmo sobrepostas.

Como metade do arquivo de entrada da rede neural admite variáveis com essas características (pisos superiores admitem 2 *APs* localizados nesse piso e outros dois no piso térreo, e vice-versa), era de se esperar (ao menos até o instante anterior à execução das simulações) que, estas variáveis influenciariam de maneira negativa no desempenho da RNA uma vez que tratam-se de ruído inserido na rede. Os resultados demonstrados na próxima seção provam o contrário.

Sendo assim, decidimos inicialmente representar graficamente o intervalo das medidas para cada ponto de acesso localizado no andar diferente àquele em análise e em seguida, apresentar o comportamento das medidas no mesmo piso. Os resultados encontram-se nas figuras 5.1-5.4:

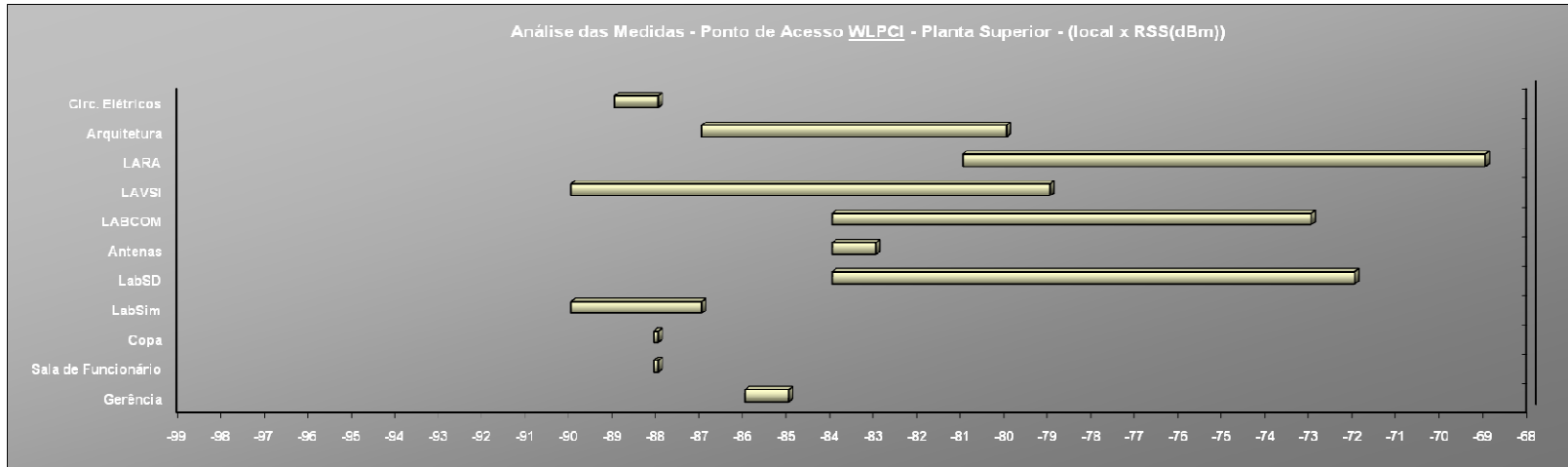


Figura 5.1 - Ponto de Acesso “WLPCI” (localizado no térreo)– Planta Superior SG-11

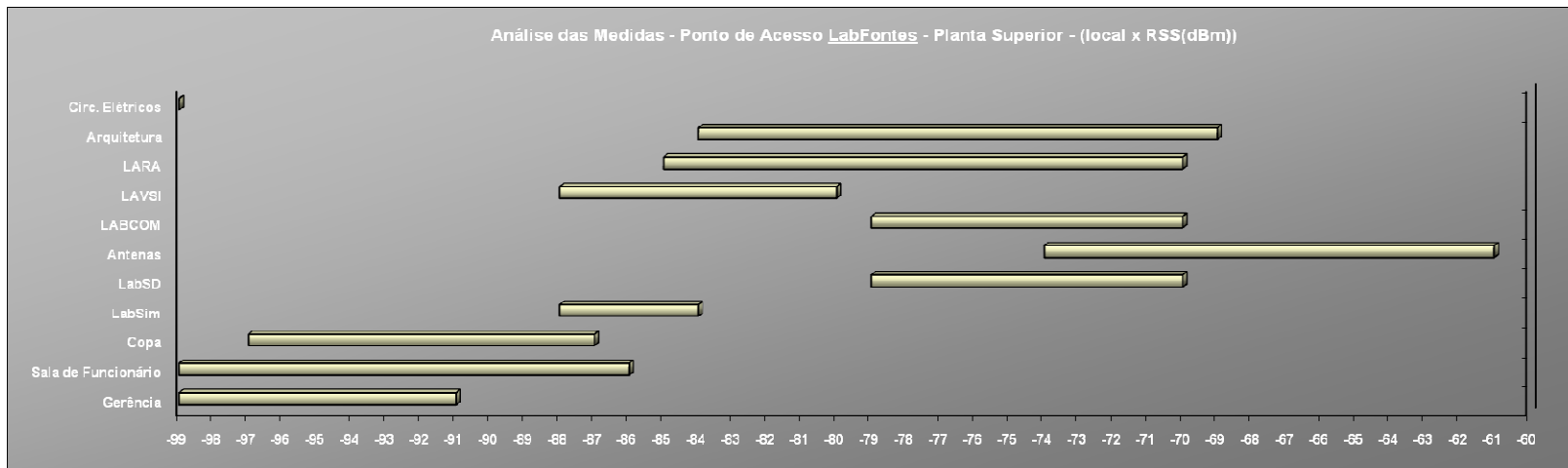


Figura 5.2 - Ponto de Acesso “LabFontes”(localizado no térreo) – Planta Superior SG-11

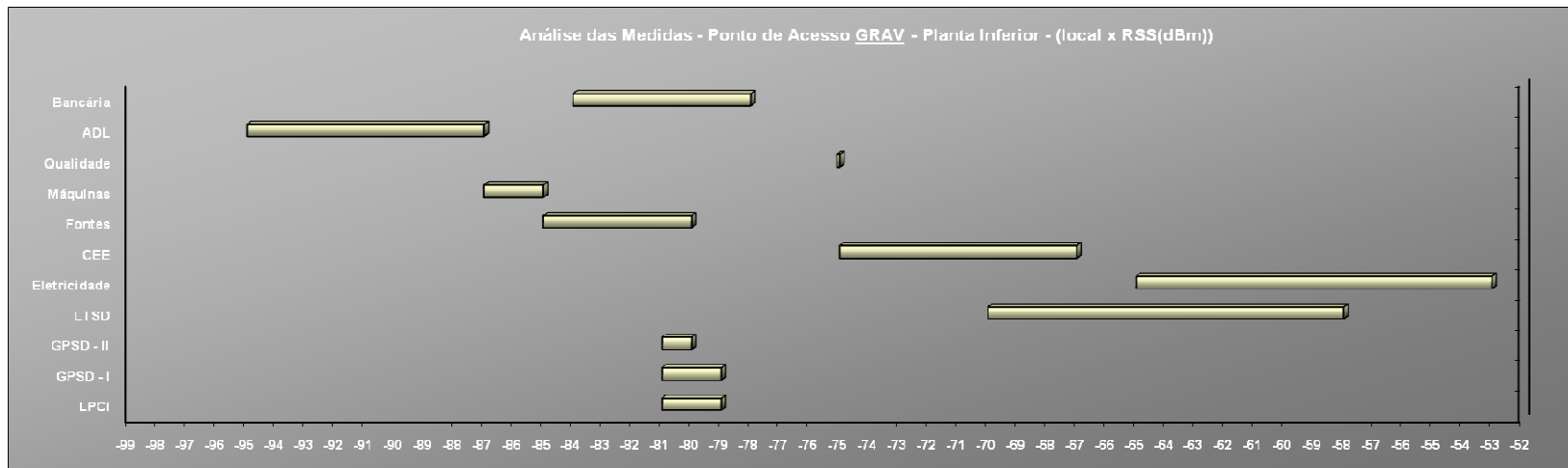


Figura 5.3 - Ponto de Acesso “GRAV”(localizado no piso superior) – Planta Inferior SG-11

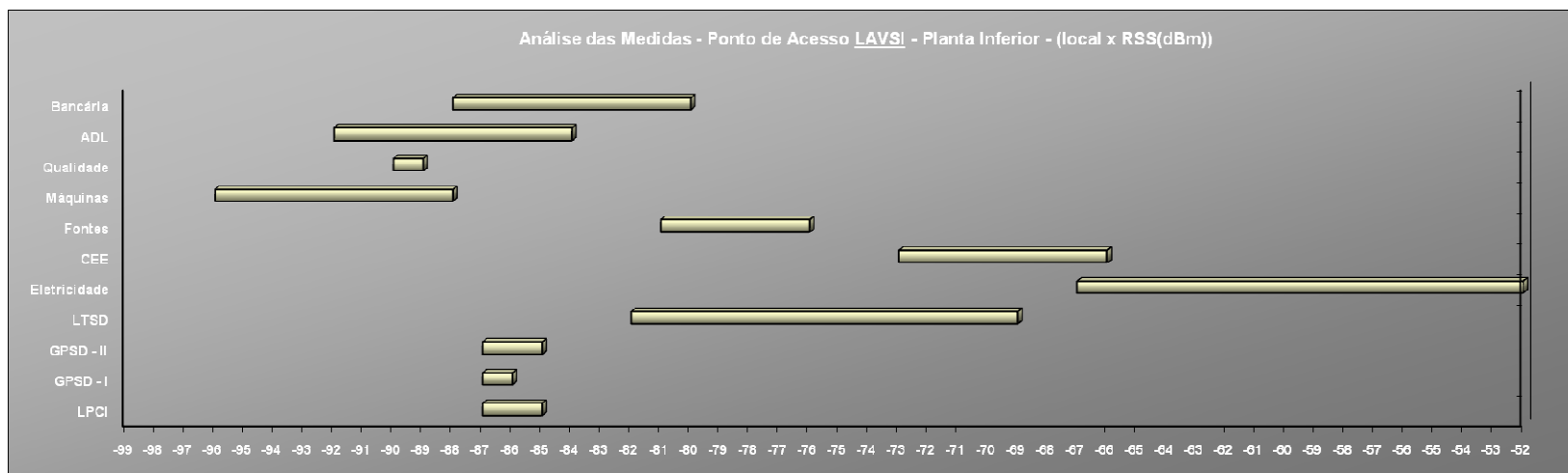


Figura 5.4 - Análise das Medidas - Ponto de Acesso “LAVSI” (localizado no piso superior)– Planta Inferior SG-11

Após analisar os intervalos das medidas concluímos que, o algoritmo de Kohonen teria um grande trabalho para separar os locais em classes distintas, uma vez que, por conta da baixa intensidade do sinal, atenuação causada pelos pisos e outros fatores inerentes ao sinal adquirido em andar diferente, tivemos grandes grupos de medidas com intervalos bastante próximos, diversas interseções e até mesmo classes dentro umas das outras.

Em seguida, exibimos algumas medidas de *APs* localizados no mesmo andar da planta em estudo. A diferença, comparando-se às medidas em andares diferentes, é perceptível. Os grupos de medidas encontram-se bem separados (características distintas), algumas com poucos pontos dentro de uma mesma classe, mas a maioria bem delineada.

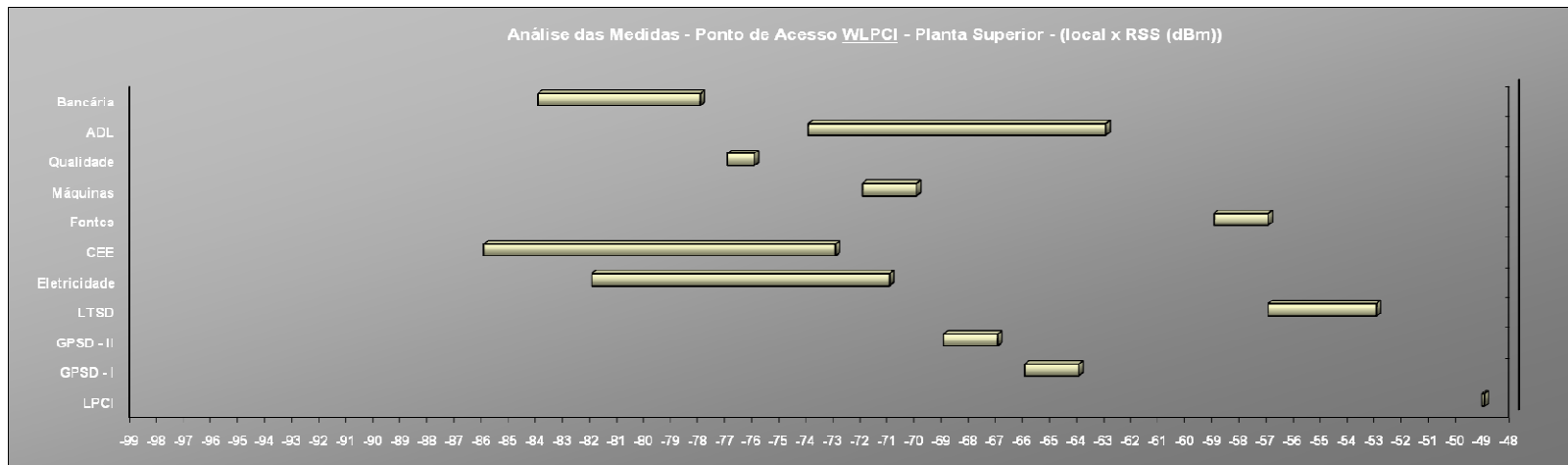


Figura 5.5 - Emissor WLPCI - Localizado no mesmo andar – Planta Inferior

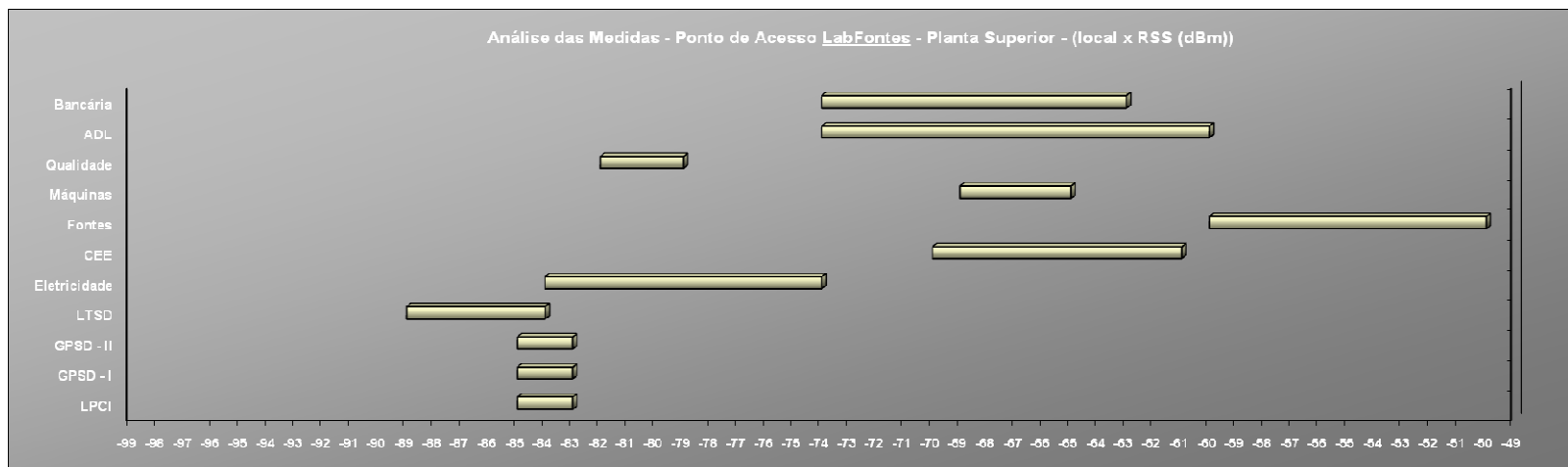


Figura 5.6 - Emissor LabFontes – Localizado no mesmo andar – Planta Inferior

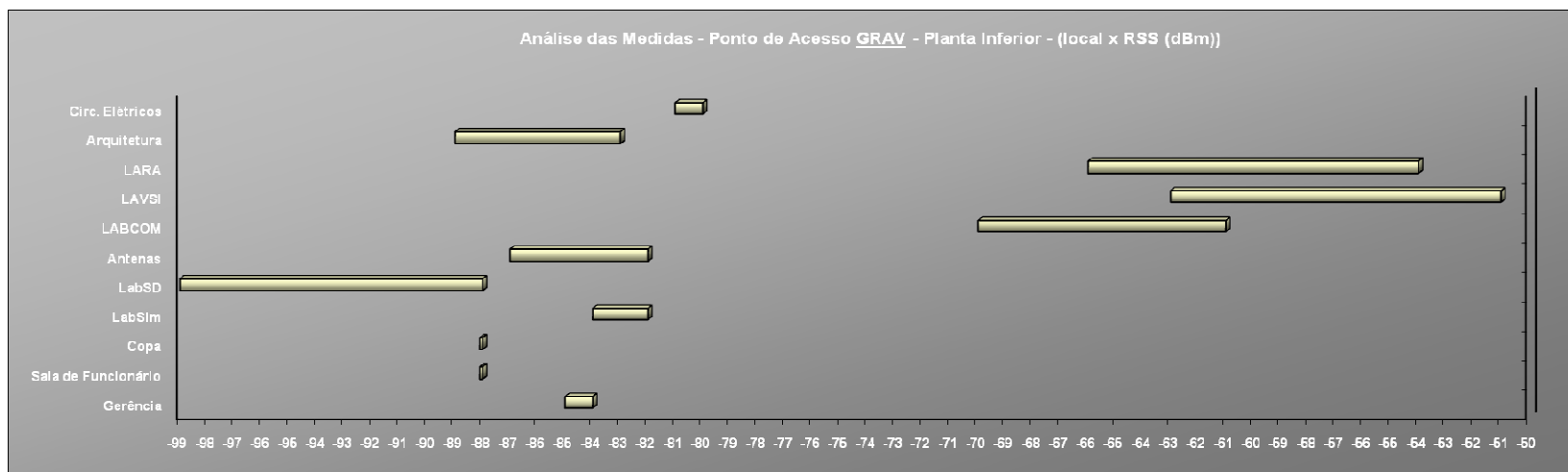


Figura 5.7 - Emissor GRAV – Localizado no mesmo andar – Planta Superior

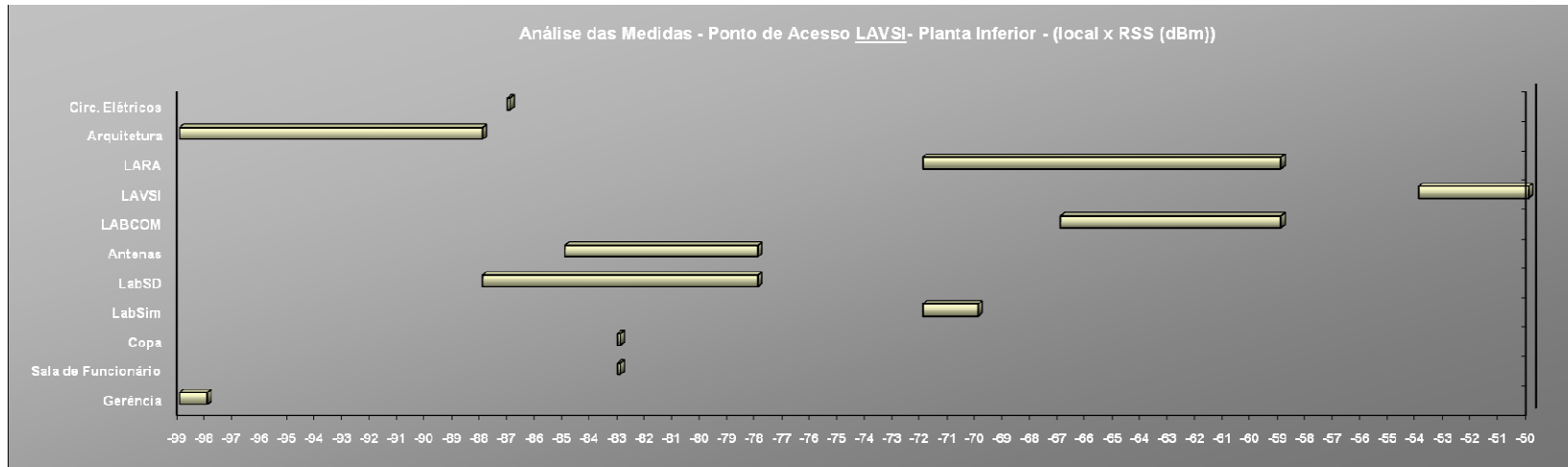


Figura 5.8 - Emissor LAVSI – Localizado no mesmo andar – Planta Superior

Após esta análise preliminar, os treinamentos foram executados com todas as 4 medidas para as duas plantas. Os resultados estão descritos a seguir.

### **5.1.2 - Resultados dos Treinamentos**

Em todos os treinamentos, o objetivo é verificar, ao final de cada iteração, o comportamento do mapa de Kohonen no que tange ao agrupamento de padrões semelhantes de medidas de *RSSI*. Isto é, se o algoritmo é capaz de identificar padrões significativos conforme a planta-baixa do ambiente.

Essa verificação é possível graças ao pós-processamento dos dados de saída da rede e uma consolidação posterior dos dados numéricos de saída com a planta-baixa do local em análise. Este resultado será exibido nas seções que se seguem.

#### 5.1.2.1 - Treinamento com a Planta Superior – SG11

Iniciou-se o treinamento da planta superior com os parâmetros especificados na seção de Definição dos Parâmetros para Treino partindo de 100 até 5.000 iterações.

A evolução do comportamento (exibindo os melhores resultados de cada iteração) da rede ao longo das iterações para cada planta encontra-se nas figuras que se seguem.



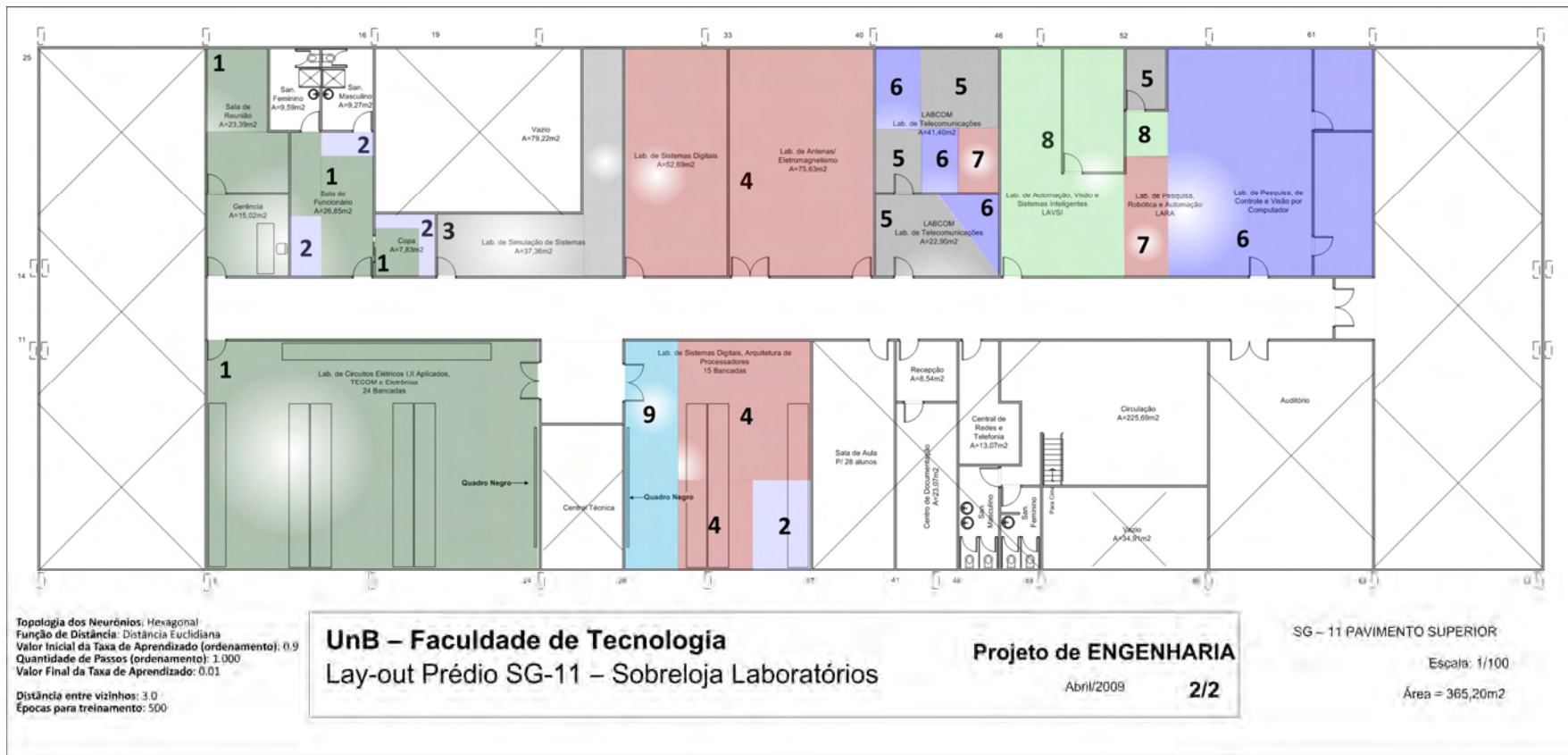


Figura 5.9 – Consolidação dos Agrupamentos em 500 iterações dos Padrões encontrados na Planta Superior – SG11

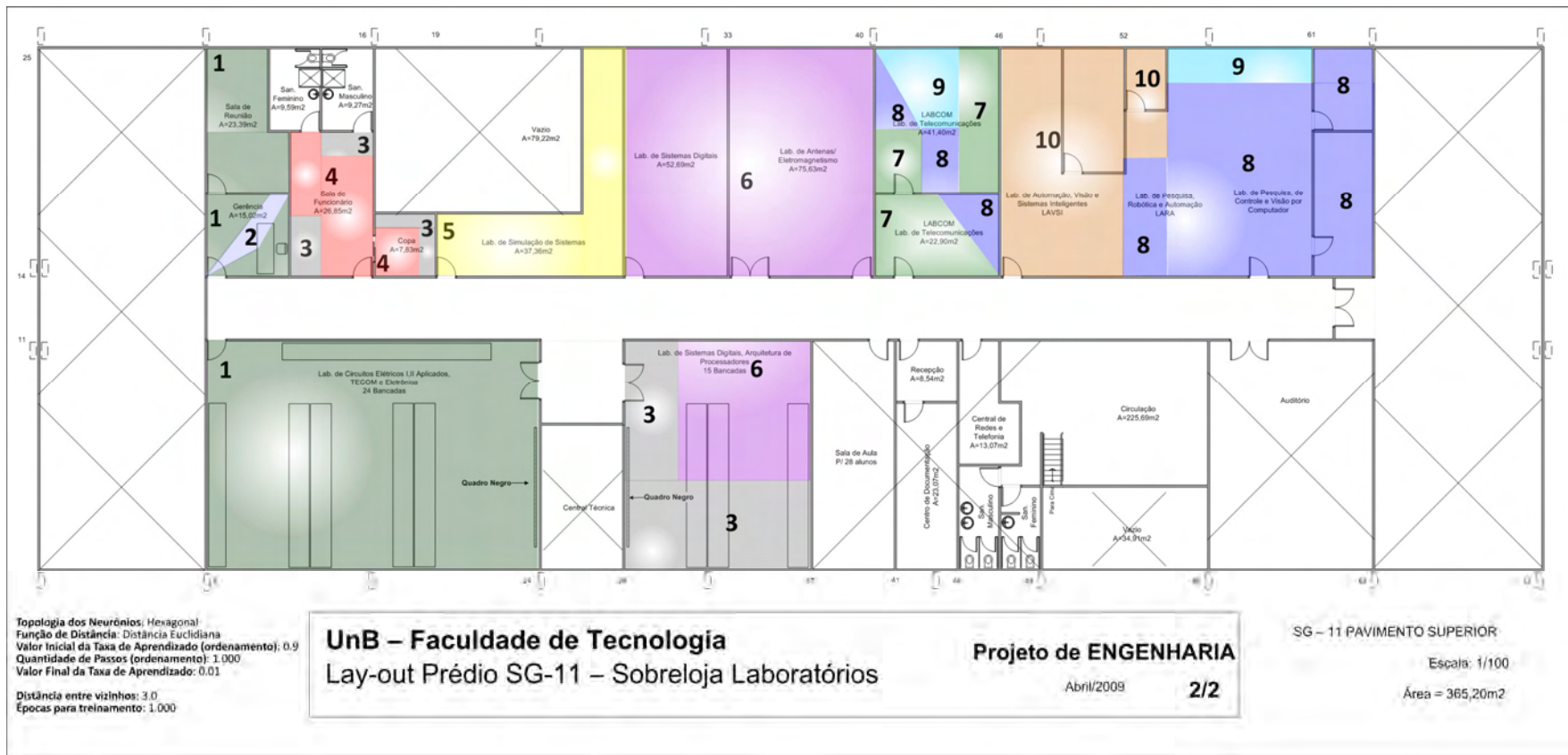


Figura 5.10 - Consolidação dos Agrupamentos em 1.000 iterações dos Padrões encontrados na Planta Superior – SG11

Por fim, os resultados finais dos agrupamentos de padrões pelo total de medidas apresentadas e a consolidação dos agrupamentos na planta-baixa do andar superior do SG-11. Estes resultados refletem a melhor configuração encontrada para a rede, portanto, configuração adequada para as futuras simulações (Figuras 5.11 e 5.12).

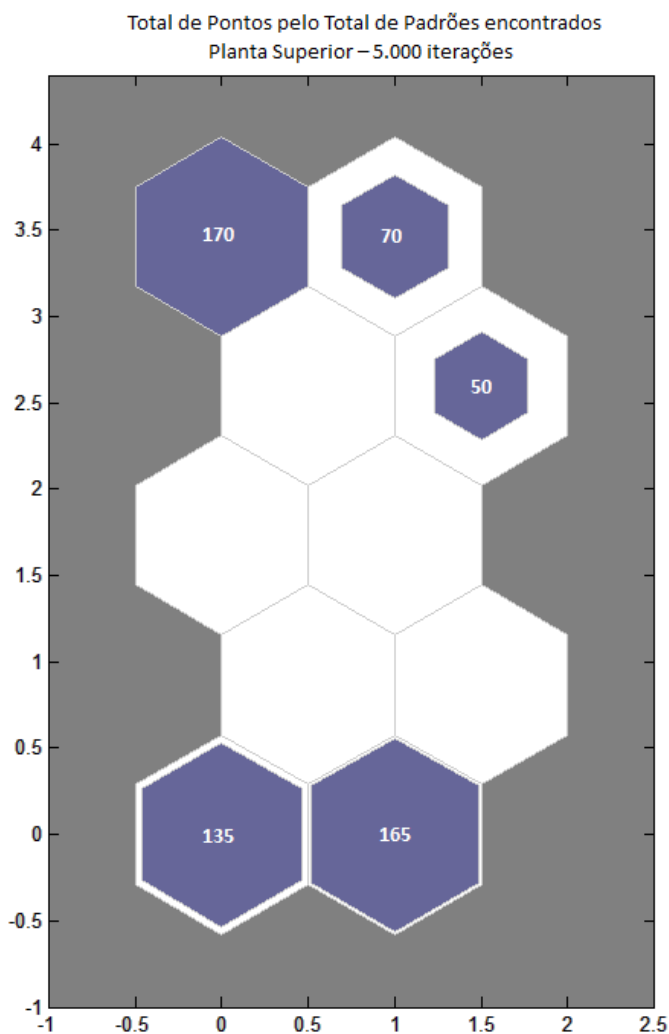


Figura 5.11 - Total de Pontos pelo Total de Padrões encontrados – Planta Superior – 5.000 iterações

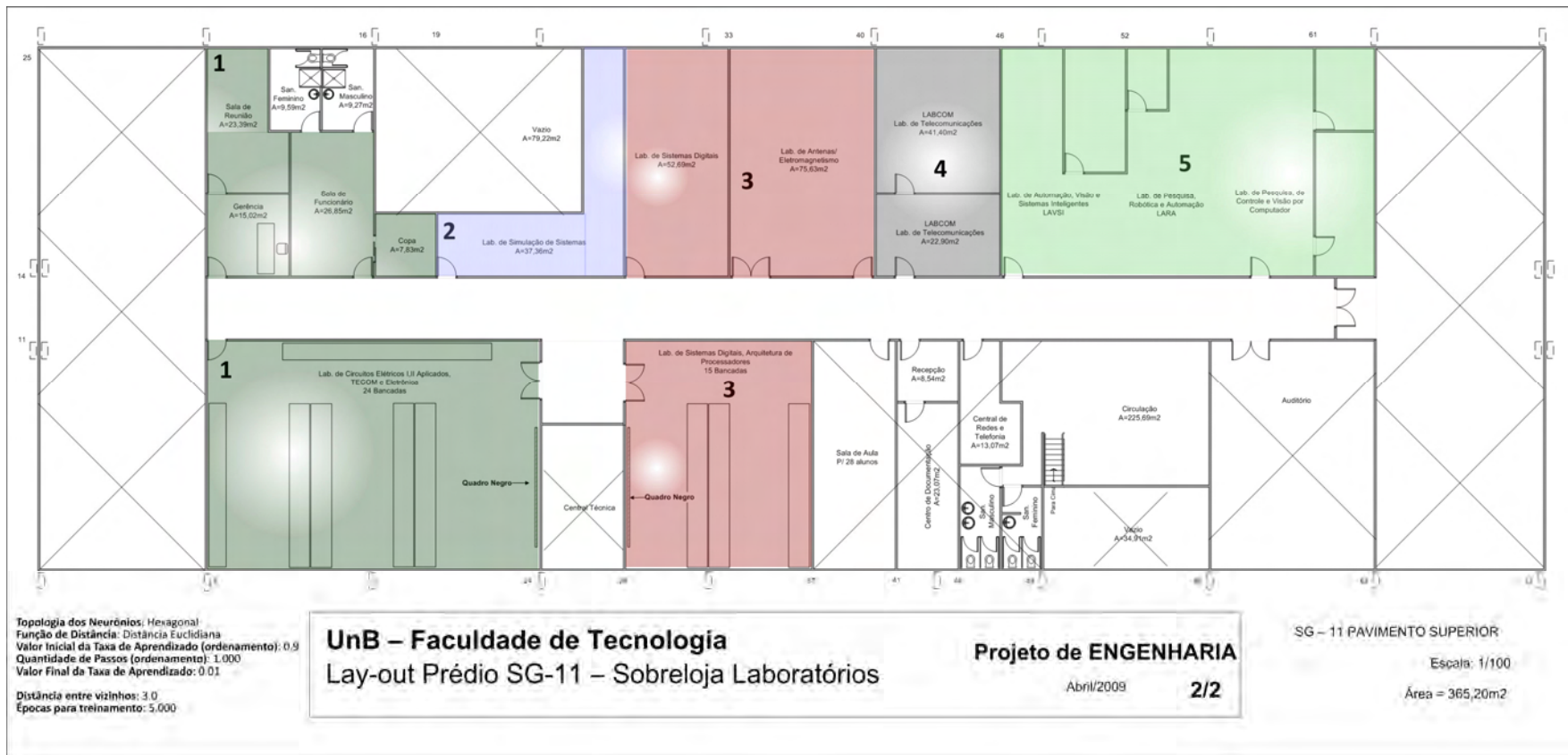


Figura 5.12 - Consolidação dos Agrupamentos dos Padrões encontrados em 5.000 iterações na Planta Superior – SG11

### 5.1.2.2 - Treinamento com a Planta Inferior – SG11

Como segunda parte do treinamento e seguindo a mesma sistemática dos testes realizados com a planta superior, iniciou-se o treinamento com os parâmetros especificados na seção de Definição dos Parâmetros para Treino para a planta inferior partindo de 100 iterações. Os resultados encontram-se nas figuras 5.13 a 5.15.

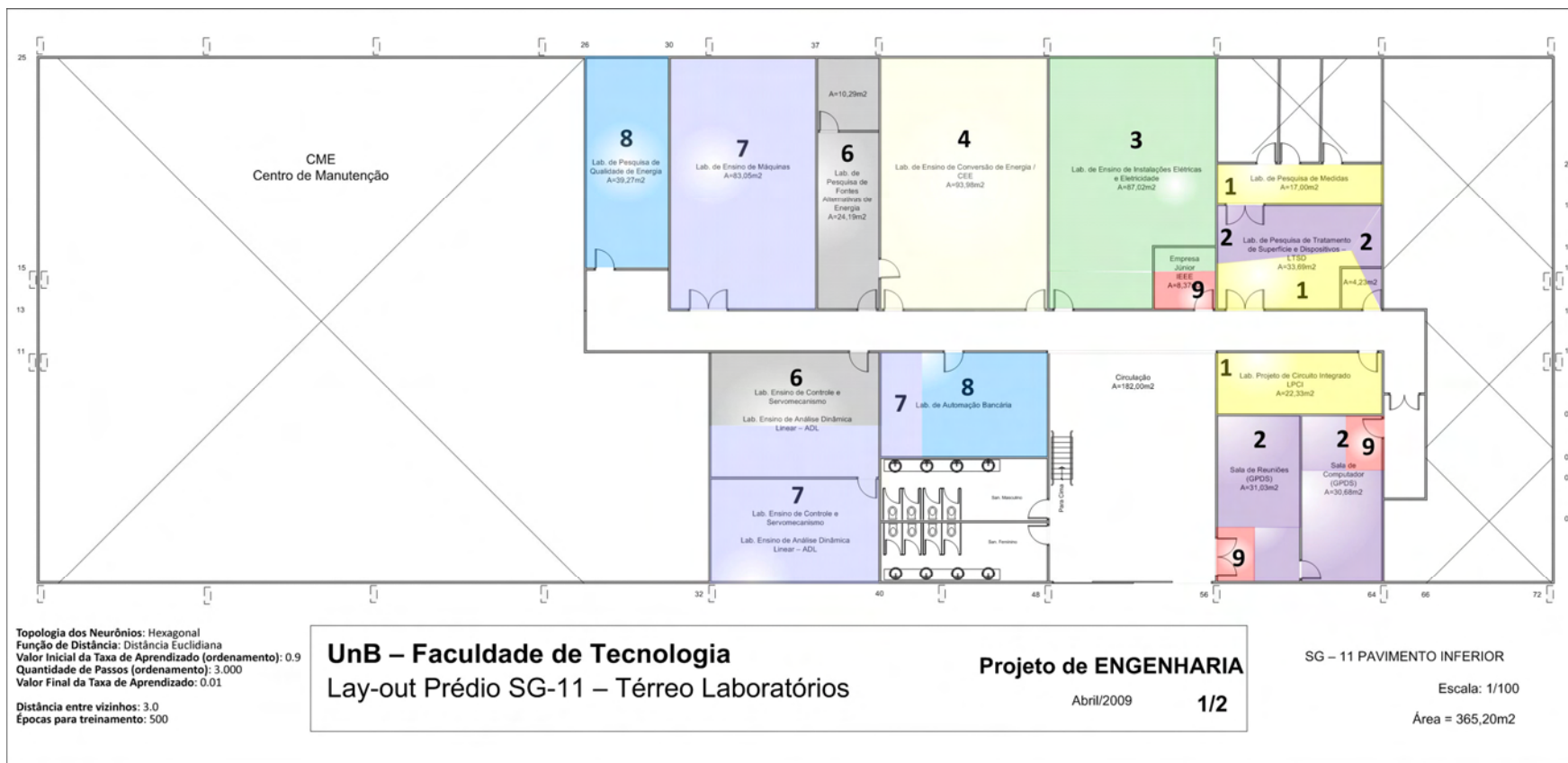


Figura 5.13 - Consolidação dos Agrupamentos em 500 iterações dos Padrões encontrados na Planta Inferior – SG11

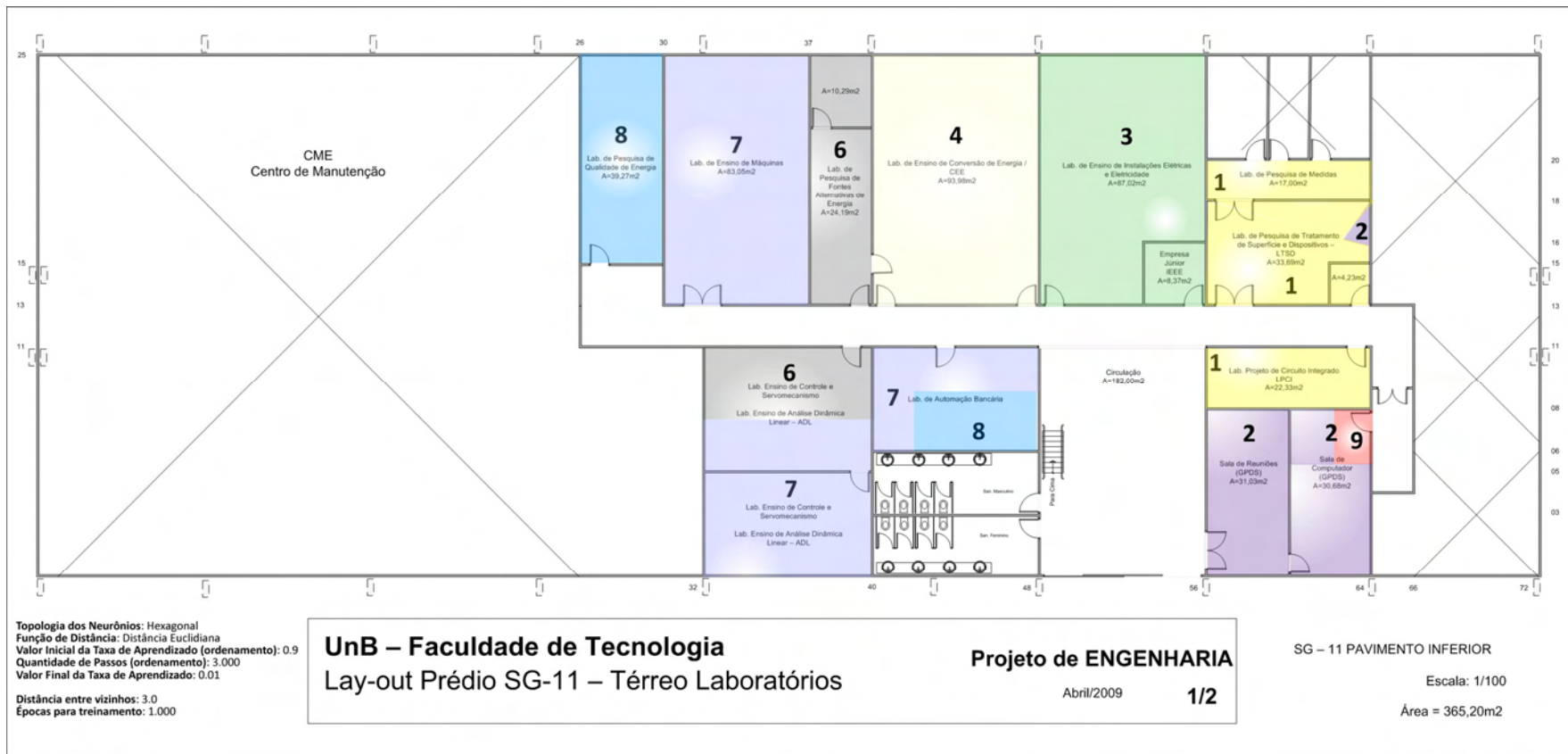


Figura 5.14 - Consolidação dos Agrupamentos em 1.000 iterações dos Padrões encontrados na Planta Inferior – SG11

Por fim, os resultados dos agrupamentos de padrões pelo total de medidas apresentadas e a consolidação dos agrupamentos na planta-baixa do andar inferior do SG-11. Estes resultados refletem a melhor configuração encontrada para a rede, portanto, configuração adequada para as futuras simulações (Figuras 5.15 e 5.16).

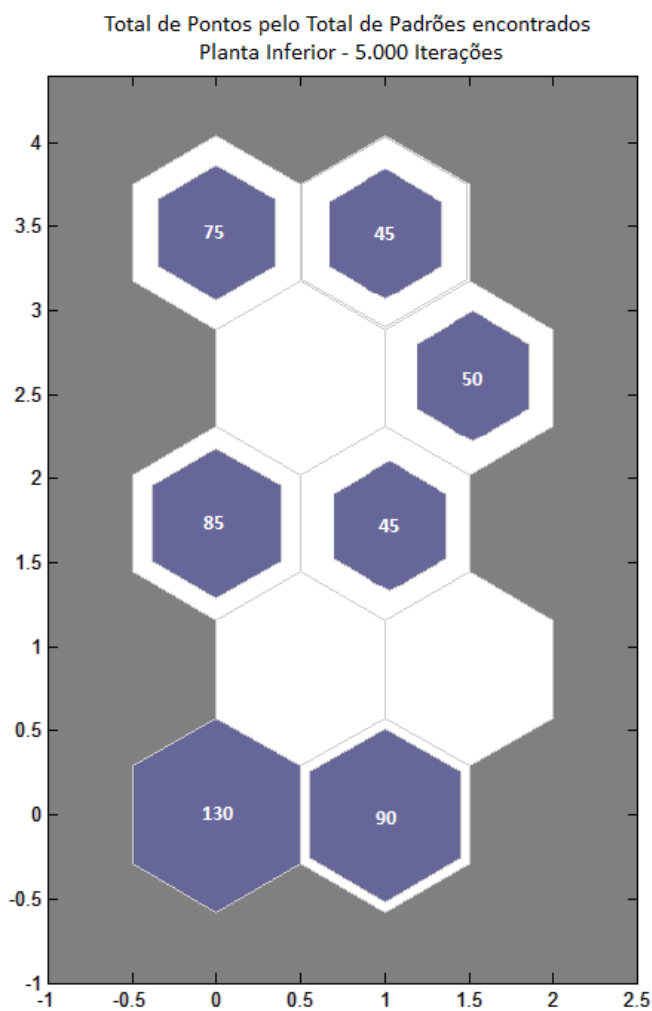


Figura 5.15 - Total de Pontos pelo Total de Padrões encontrados – Planta Inferior – 5.000 iterações



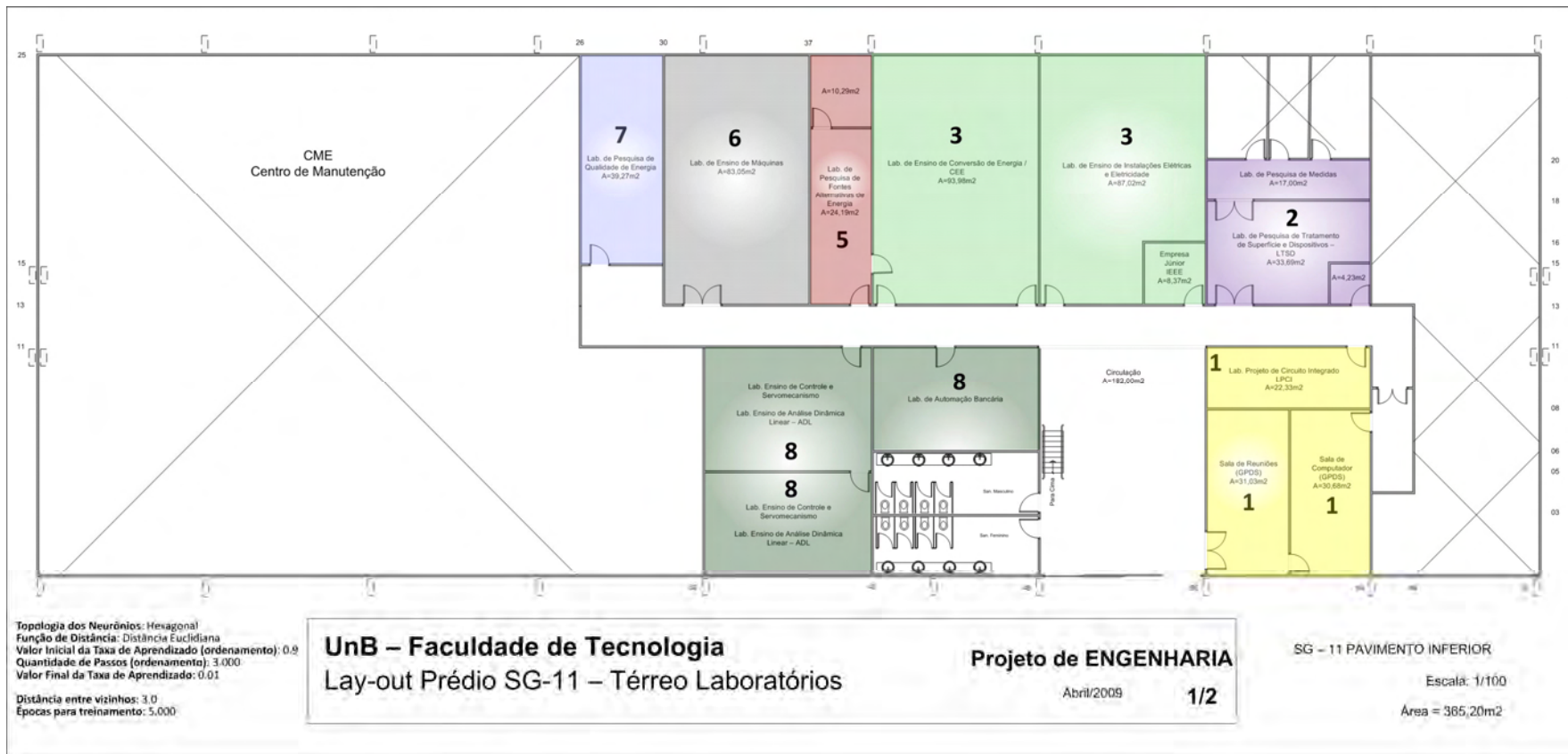


Figura 5.16 - Consolidação dos Agrupamentos dos Padrões encontrados em 5.000 iterações na Planta Inferior – SG11

### 5.1.2.3 - Treinamento com a Planta Completa – SG11

O treinamento contendo os dados da planta completa do SG-11 apresentou resultados semelhantes àqueles encontrados separadamente nas plantas Superior e Inferior. Os resultados de todos os treinamentos são discutidos nas seções a seguir. As figuras 5.17 a 5.19 ilustram o resultado final do Mapa resultante do treinamento da Rede Neural de Kohonen para as duas plantas utilizadas no experimento.

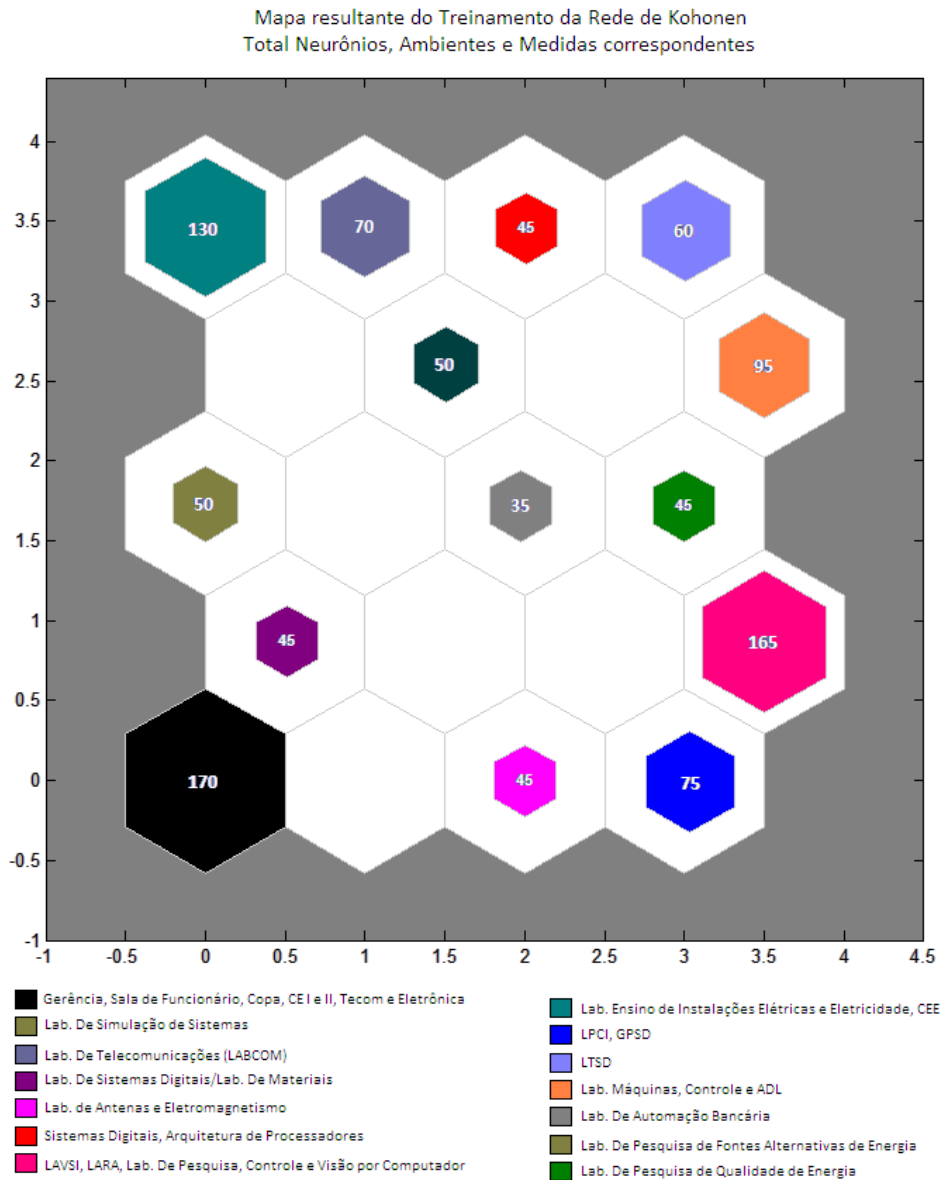


Figura 5.17 - Mapa Resultante do Treinamento da Rede Neural de Kohonen – Planta Completa – 10.000 iterações

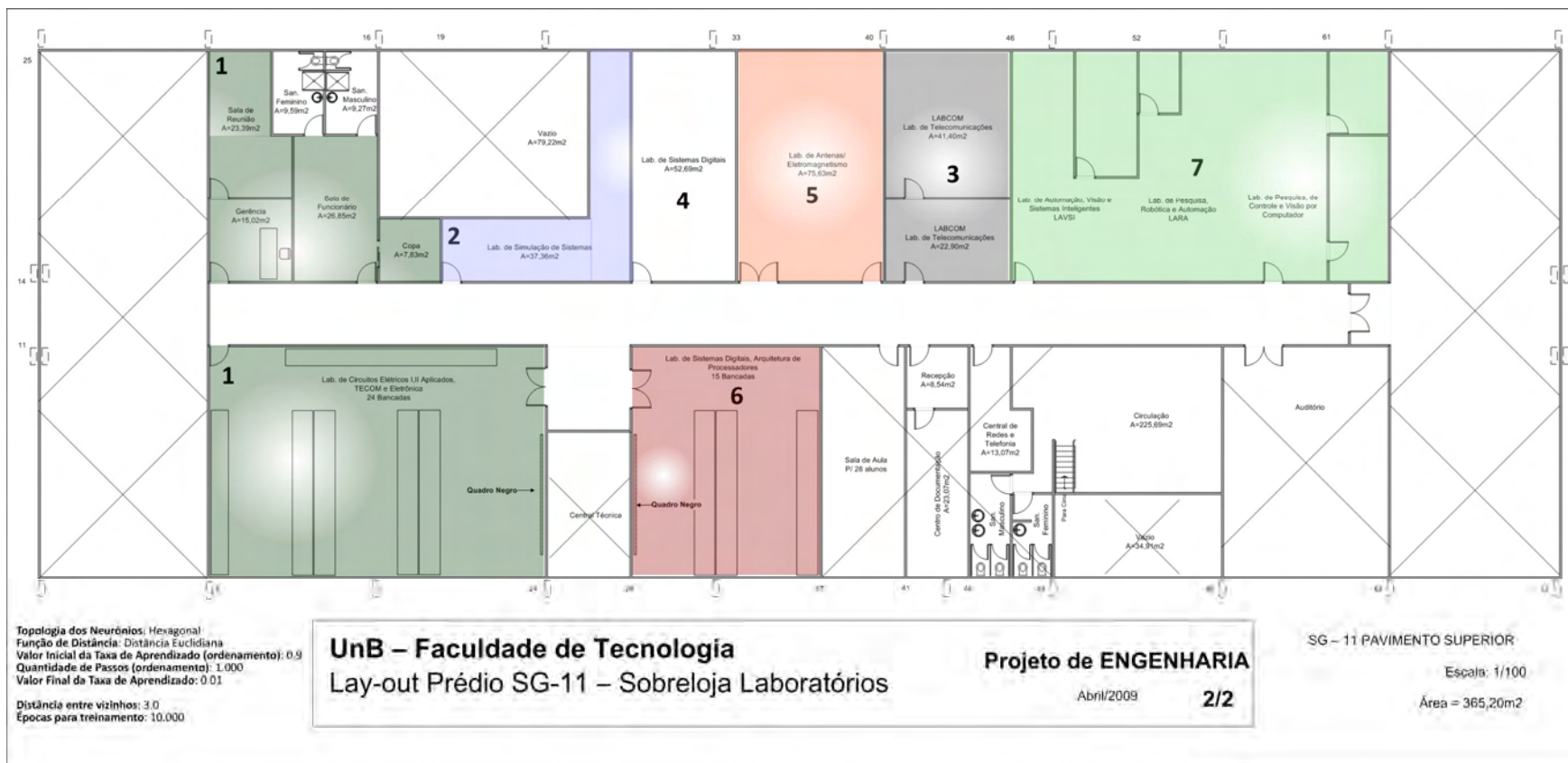


Figura 5.18 - Consolidação dos Agrupamentos dos Padrões encontrados em 10.000 iterações na Planta Completa (Superior) – SG11

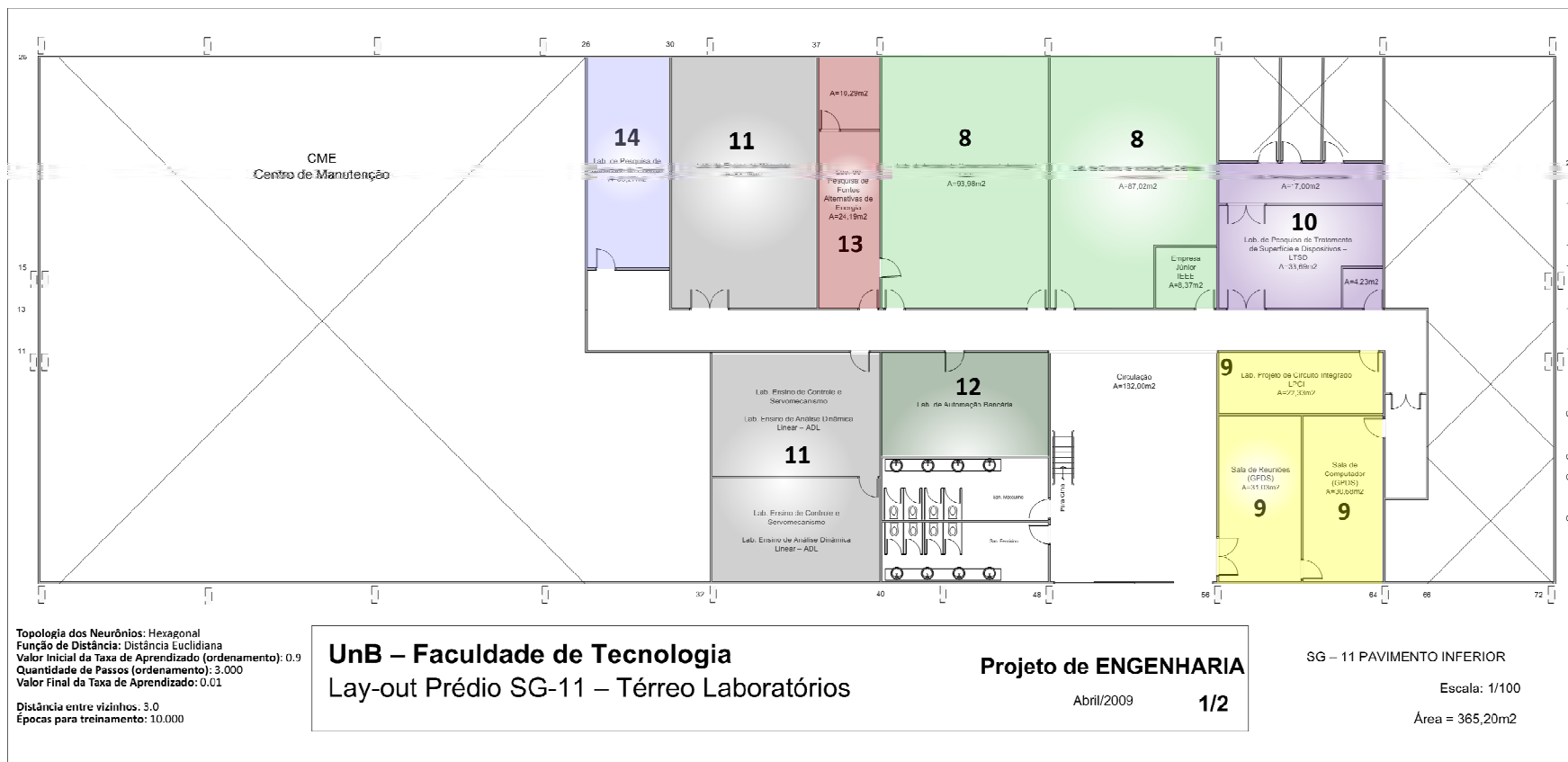


Figura 5.19 - Consolidação dos Agrupamentos dos Padrões encontrados em 10.000 iterações na Planta Completa (Inferior)– SG11

#### 5.1.2.4 - Considerações a respeito da fase de Treinamento

Por meio da análise das figuras apresentadas na seção anterior, a primeira influência que podemos perceber nitidamente é a **quantidade de iterações** na qual a rede neural é submetida. Percebemos que, à medida que os dados de entrada são apresentados a rede, melhor ela é capaz de identificar padrões significativos e agrupá-los. Então, em um treinamento com poucas iterações, possivelmente teremos pontos dentro de um mesmo ambiente em classes distintas (o que é um resultado indesejável para o problema). Nesse caso, existem dois neurônios respondendo por um mesmo ambiente na planta. À medida que o treinamento alcança um número maior de iterações, os mesmos pontos, que em treinamentos anterior estavam em classes distintas, passam a compor a classe dominante do ambiente. Assim, temos o resultado final e desejável da rede, que é um único neurônio respondendo por todas as medidas do ambiente (laboratório ou sala de aula).

Essa característica só nos leva a confirmar as orientações de Kohonen e de diversos estudos que dizem que a quantidade de iterações está relacionada com a quantidade de neurônios da rede. Sendo assim, uma rede com 20 neurônios deve passar, em média, por 500 vezes o total de neurônios, resultando, assim, em 10.000 iterações (caso da planta completa). Observe que, seguindo esse entendimento, nas iterações próximas ao valor recomendado, o mapa está praticamente coerente com o que é esperado (a maioria dos pontos dentro de um mesmo ambiente pertencente ao único neurônio vencedor da competição naquela região).

Outra observação importante é relativa à **distância da vizinhança**. Nos treinamentos executados, essa distância deve ser realmente grande e ser reduzida com o passar das iterações. Em simulações iniciadas com distâncias na ordem de 1 ou 1,5, não encontramos resultados consistentes. Por este motivo, adotou-se a distância da ordem de 3 com decréscimos ao longo do tempo, para, ao final, encontramos distâncias entre 1,0 e 1,5 (figura 5.20).

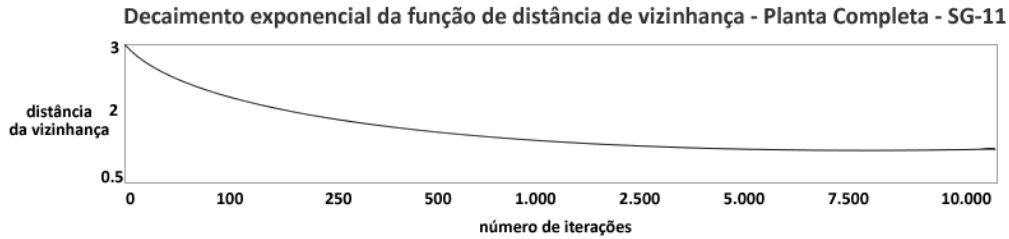


Figura 5.20 - Decaimento exponencial da função de distância de vizinhança – Planta Completa – SG-11

A quantidade de **passos de ordenamento** é também item importante e merece algumas considerações. A bibliografia especializada recomenda valores próximos ou iguais a 1.000. Fizemos diversas simulações variando este parâmetro para valores menores e maiores que o recomendado, caso em que obtivemos resultados pouco consistentes. O valor padrão nos trouxe resultados mais significativos, por este motivo, adotado para as futuras simulações.

A mesma consideração é válida para a **taxa de aprendizado** na fase de ordenamento e convergência. Utilizamos valores na faixa de 0,9 a 0,01 (figura 5.21). A rede apresentou melhores resultados quando utilizamos estes valores. Também foram realizadas simulações com valores entre 0,5 a 0,001, inclusive o valor 0 (indo de encontro as recomendações dos estudiosos de redes neurais) e constatamos que valores neste intervalo são prejudiciais à rede .

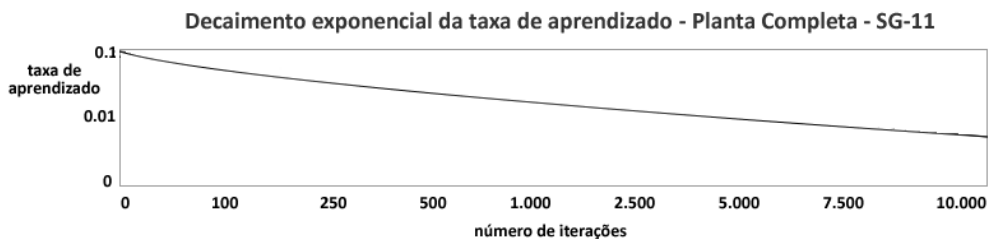


Figura 5.21 - Decaimento exponencial da taxa de aprendizado – Planta Completa – SG-

11

Sendo assim, após diversas combinações de variáveis e simulações executadas, os parâmetros que apresentaram os melhores resultados para cada tipo de simulação são os seguintes (Tabela 5.1):

Tabela 5.1 - Parâmetros utilizados por tipo de treinamento para a RNA Kohonen

Parâmetro	Planta Superior	Planta Inferior	Planta Completa
<i>Dimensão do Mapa</i>	<b>[2 5]</b>	<b>[2 5]</b>	<b>[4 5]</b>
<i>Taxa de aprendizado (Ordenamento)</i>	<b>0,9</b>	<b>0,9</b>	<b>0,9</b>
<i>Passos de Ordenamento</i>	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>
<i>Taxa de aprendizado (Convergência)</i>	<b>0,01</b>	<b>0,01</b>	<b>0,01</b>
<i>Distância de Vizinhaça (Inicial)</i>	<b>3,0</b>	<b>3,0</b>	<b>3,0</b>
<i>Iterações</i>	<b>5.000</b>	<b>5.000</b>	<b>10.000</b>

Em se tratando do tempo de treinamento, observe os resultados na tabela 5.2:

Tabela 5.2 - Tempo de Execução na fase de Treinamento para a RNA Kohonen (no formato: h:min:s)

	Iterações						
	100	250	500	1.000	2.500	5.000	10.000
Planta Superior	00:01:53	00:05:06	00:09:29	00:19:16	00:52:58	01:28:26	--
Planta Inferior	00:01:32	00:04:26	00:07:46	00:15:32	00:44:14	01:20:18	--
Planta Completa	00:03:54	00:09:26	00:19:26	00:39:30	01:41:17	03:02:29	05:44:28

É de ressaltar, que apesar dos tempos de treinamento serem elevados para a planta completa em sua melhor configuração, devemos lembrar que a rede está lidando com milhares de medidas e processando as mesmas diversas vezes. Além disso, o tempo de execução é intimamente ligado ao equipamento empregado nos treinamentos. Por fim, a fase de treinamento não é fato corriqueiro. Na verdade, a rede é treinada uma única vez e a partir desse momento, ela estará apta para o seu uso principal (as simulações). De maneira geral, as simulações possuem respostas calculadas em segundos, tornando-se muito vantajosas em se comparado aos métodos tradicionais de localização.

### 5.1.3 - Simulação

Neste ponto, a rede neural encontra-se treinada, os parâmetros ideais foram identificados e o funcionamento da rede é estável e apresenta saídas confiáveis para esta etapa. Resta agora, iniciar a fase de simulações com o objetivo de por a prova toda a capacidade do experimento por meio da apresentação de novos cenários, cenários já conhecidos anteriormente, além de situações que exigem um processamento e precisão elevados. Sendo assim, a simulação foi dividida em diversas etapas:

1. Simulação contendo pontos utilizados no treinamento
2. Simulação contendo pontos localizados em regiões de fronteira
3. Simulação contendo novos pontos adquiridos em novas medidas de campo aleatórias

Para todas as simulações, consideramos a rede neural treinada com a planta completa. Os gráficos que avaliam os resultados de cada etapa da simulação podem contemplar erros de localização de medidas entre andares (caso existam). Do contrário, considere que o quantitativo de pontos incorretos e/ou percentual de erro se refere unicamente a erros de localização ocorridos no mesmo piso em análise.

#### 5.1.3.1 - Simulação contendo pontos utilizados no treinamento

Para esta etapa de simulação o critério adotado consistiu na seleção aleatória de 75% dessas amostras adquiridas para cada ambiente em estudo. As figuras 5.22 e 5.23 ilustram a seleção.



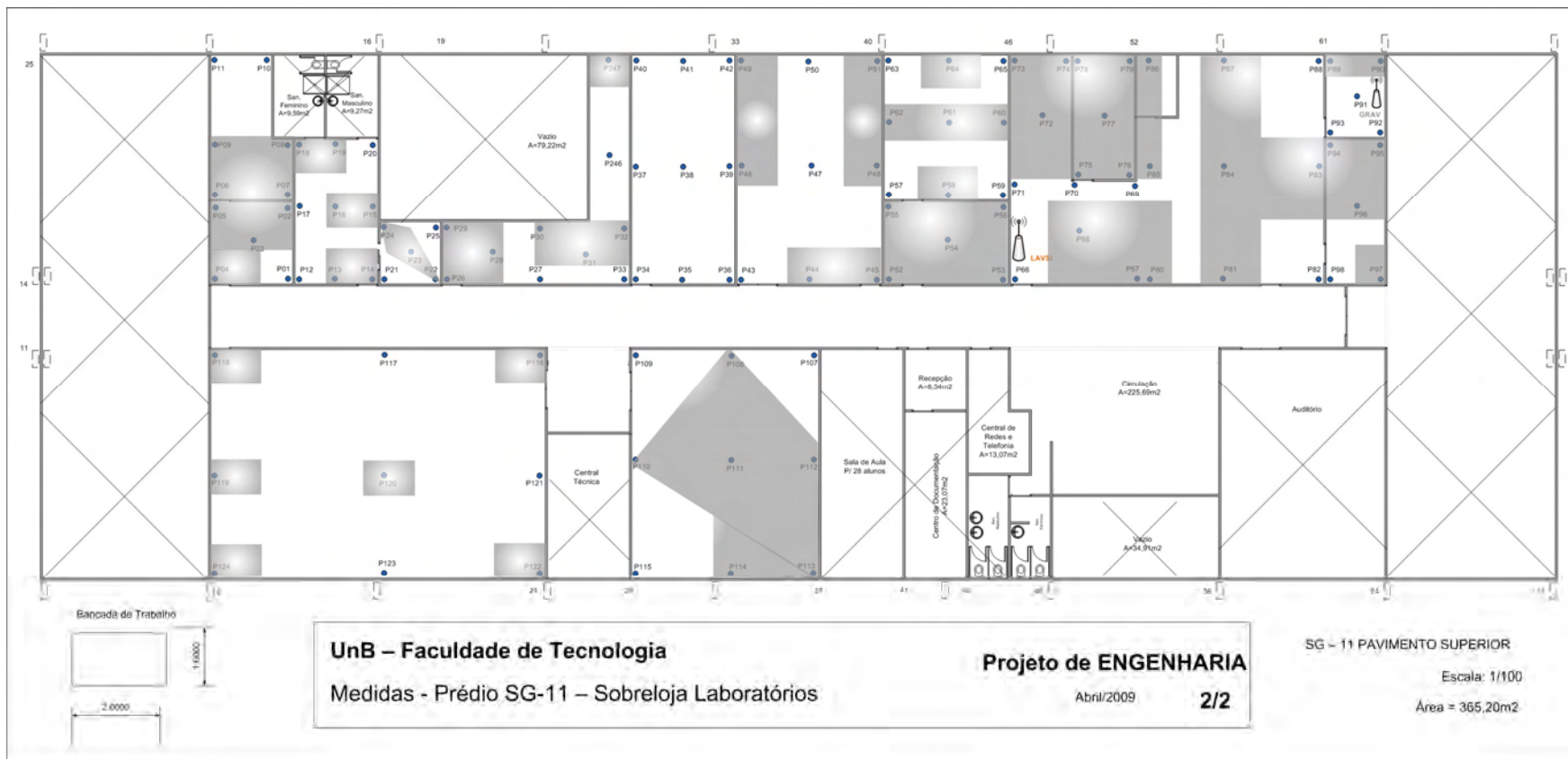


Figura 5.22 - Seleção aleatória de 75% das amostras para a Planta Superior – RNA Kohonen

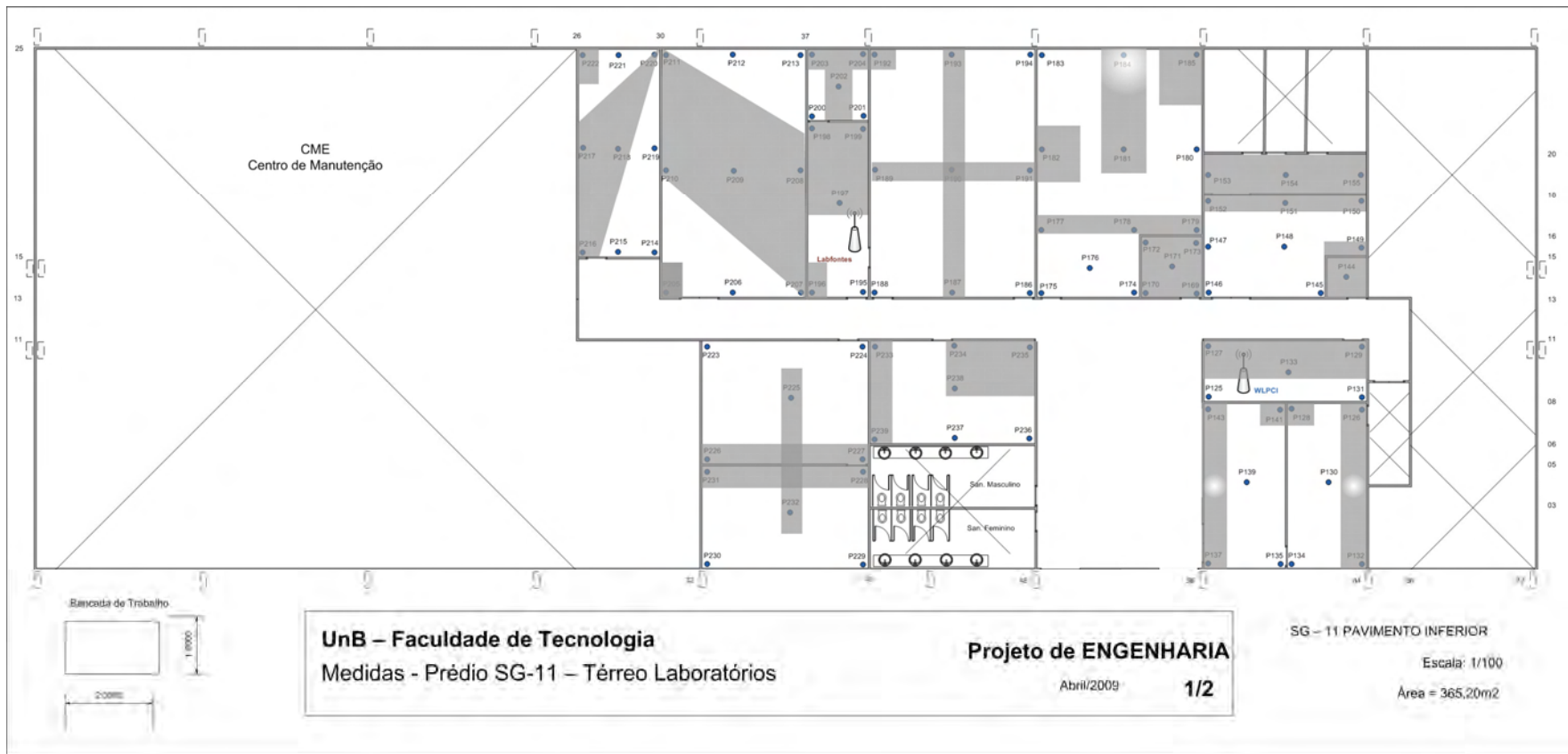


Figura 5.23 - Seleção aleatória de 75% das amostras para a Planta Inferior – RNA Kohonen

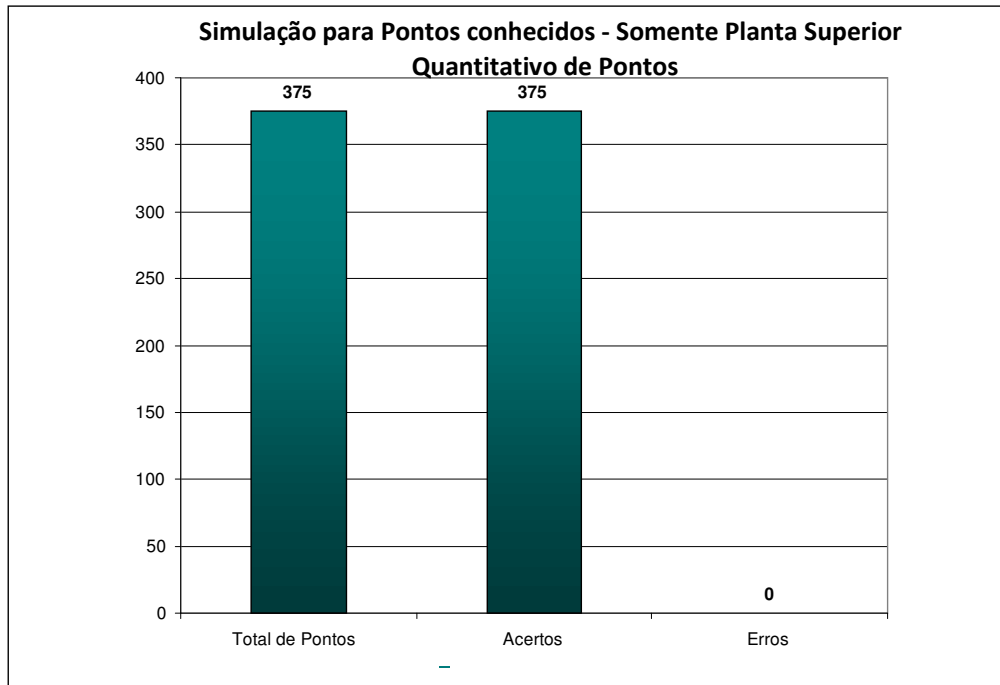


Figura 5.24 - Resultados alcançados para a Planta Superior – Quantitativos de Pontos – RNA Kohonen – 100% de Acerto

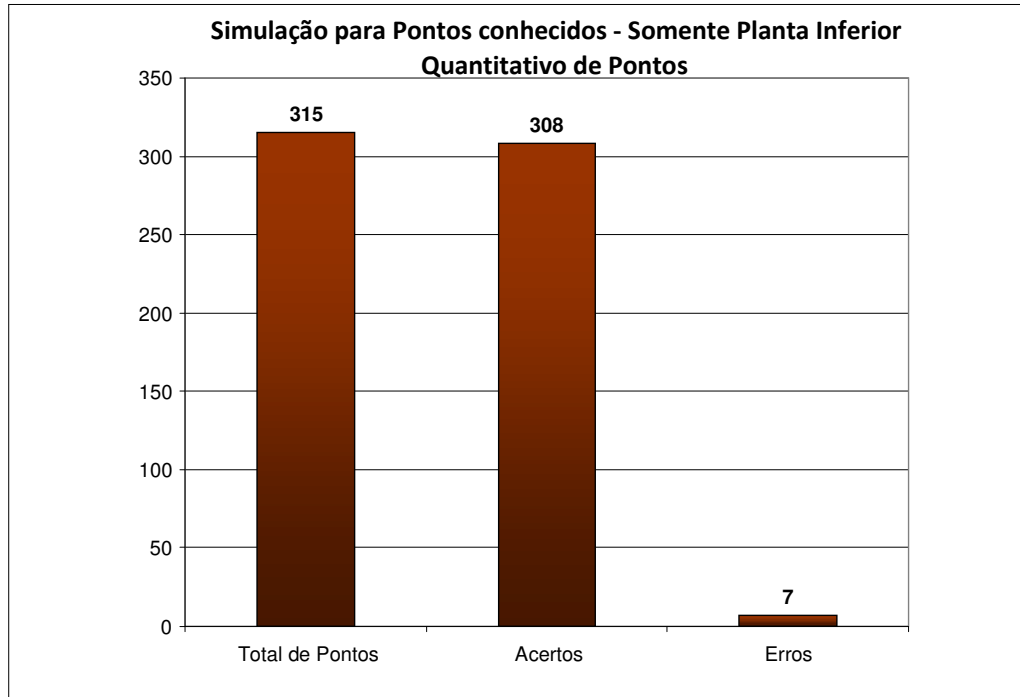


Figura 5.25 - Resultados alcançados para a Planta Inferior – Quantitativos de Pontos – RNA Kohonen – 97,78% de Acerto e 2,22% de Erro

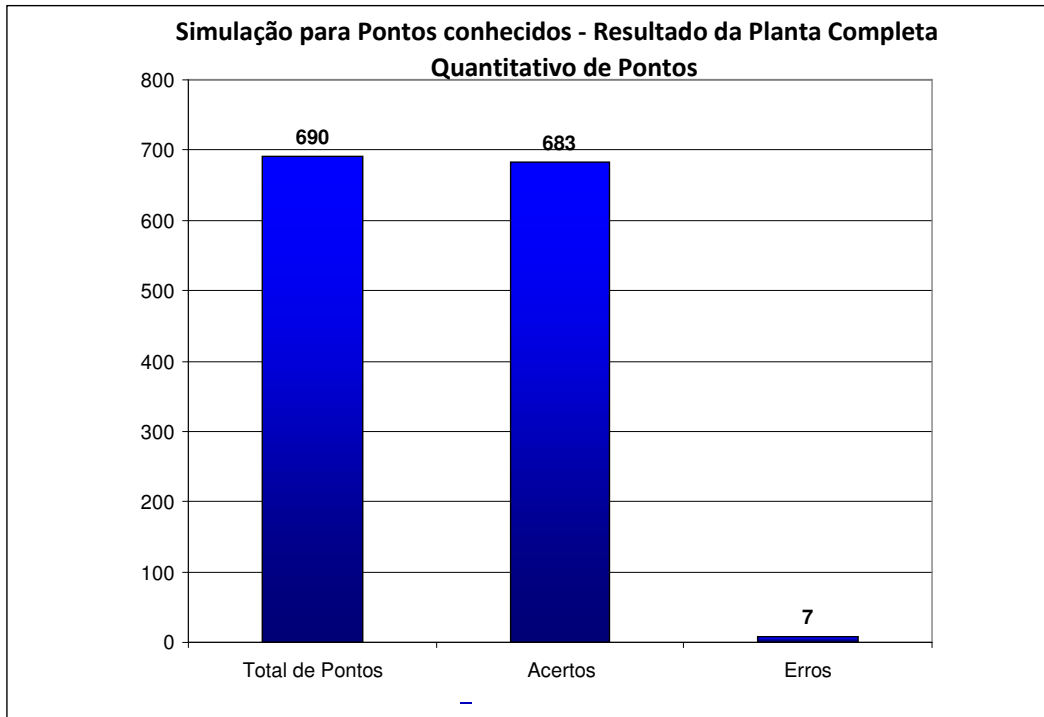


Figura 5.26 - Resultados alcançados para a Planta Completa (Resultados Consolidados)  
 – Quantitativos de Pontos – RNA Kohonen – 98,99% de Acerto e 1,01% de Erro

Os gráficos nos mostram que para pontos já conhecidos da rede, as simulações tendem a não revelar erros de reconhecimento de local. Do total de 315 pontos, a rede apresentou 1,02% de erro, ou 7 pontos fora do local. Mesmo considerando incorreta a localização, os resultados apresentados pela rede indicam pontos com localização adjacente à correta (Figuras 5.24 a 5.26). Não ocorreram erros de localização entre andares

#### 5.1.3.2 - Simulação contendo pontos localizados em regiões de fronteira

Houve uma preocupação em se verificar se o processo de localização funcionava bem em regiões de fronteira, que são as áreas entre uma classe e outra no mapa auto-organizável. Partindo deste critério, a simulação trabalhou com 270 amostras totais, espalhadas em planta superior (140 amostras) e planta inferior (130 amostras). As figuras 5.27 e 5.28 contemplam essa distribuição.

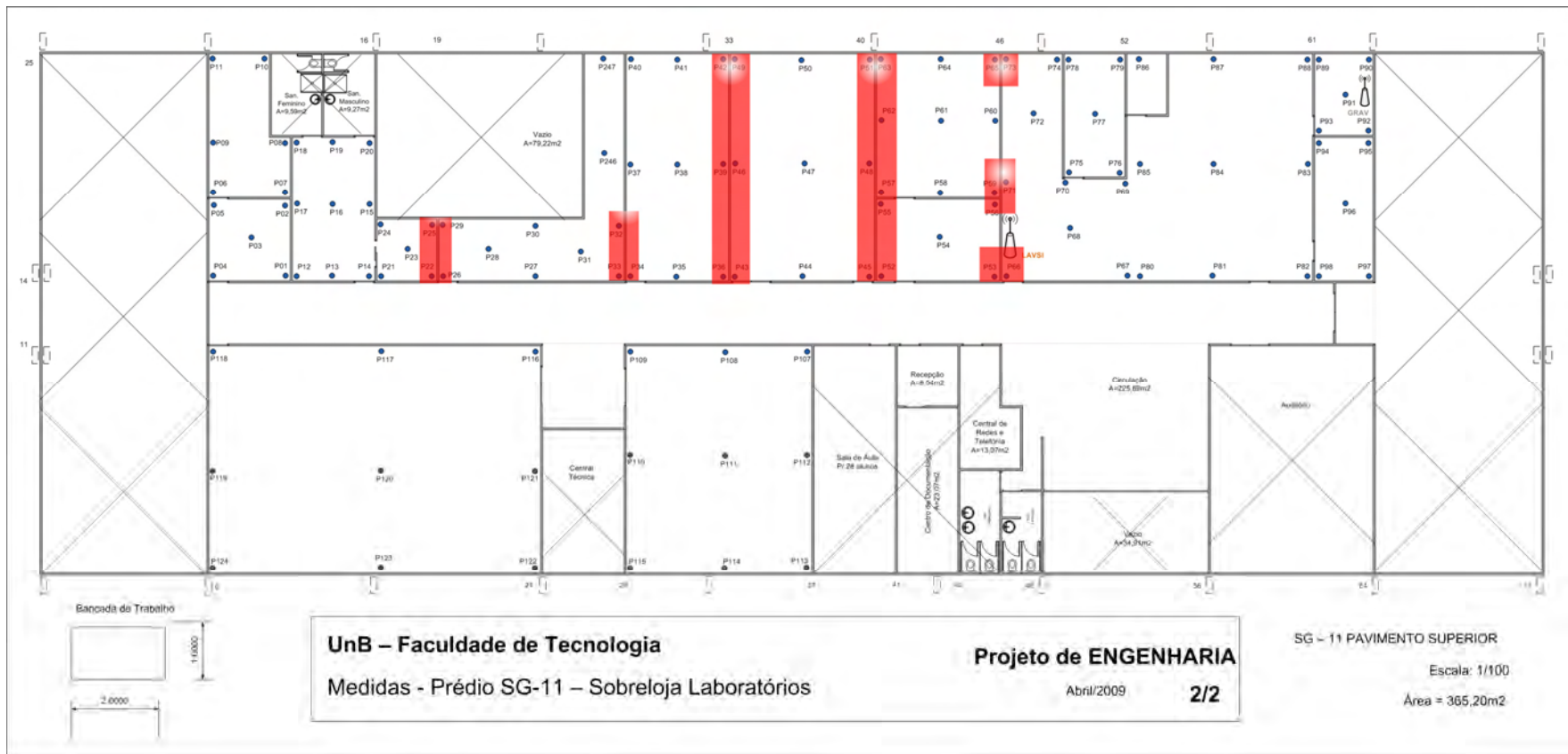


Figura 5.27 - Pontos selecionados em Regiões de Fronteira – Planta Superior – RNA Kohonen

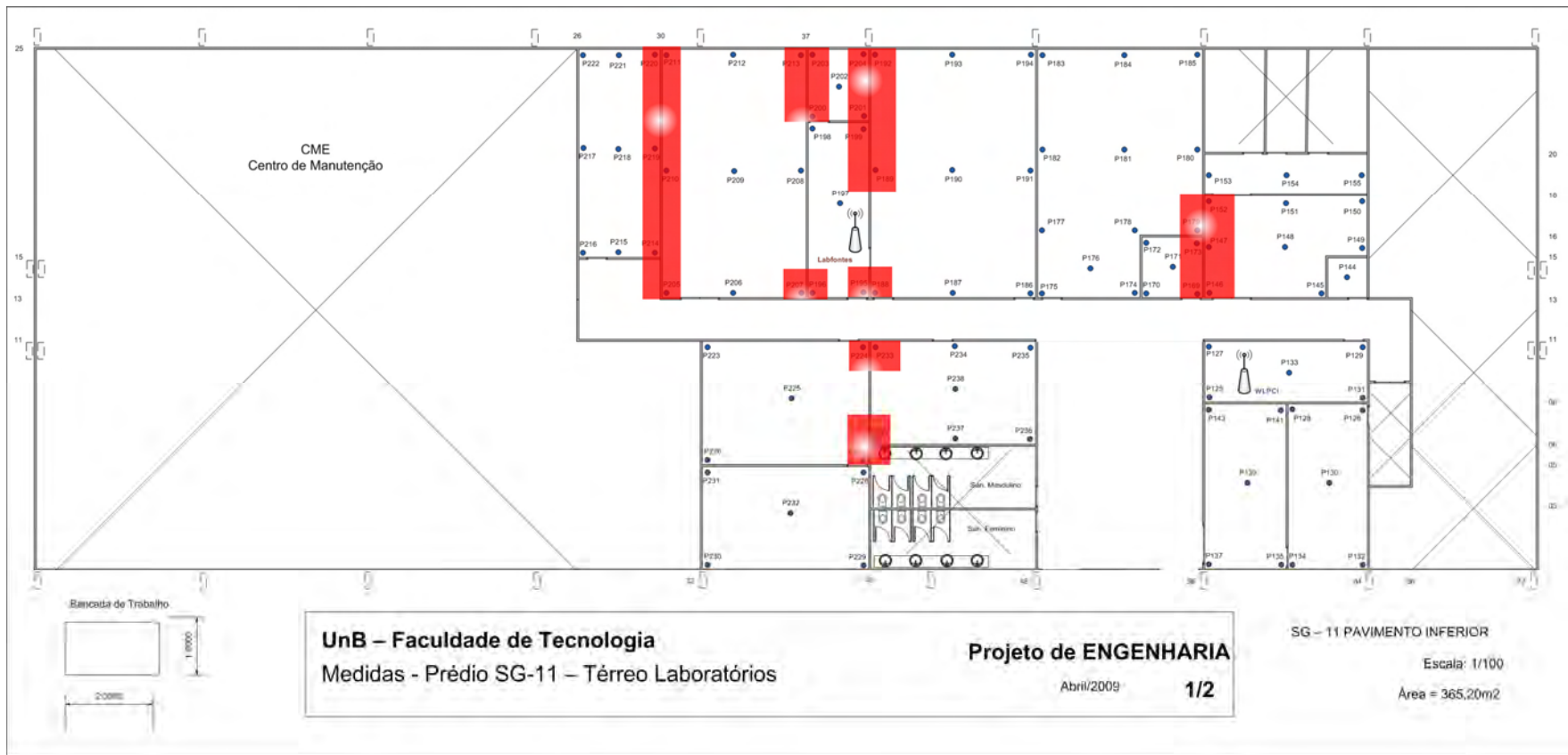


Figura 5.28 - Pontos selecionados em Regiões de Fronteira – Planta Inferior – RNA Kohonen

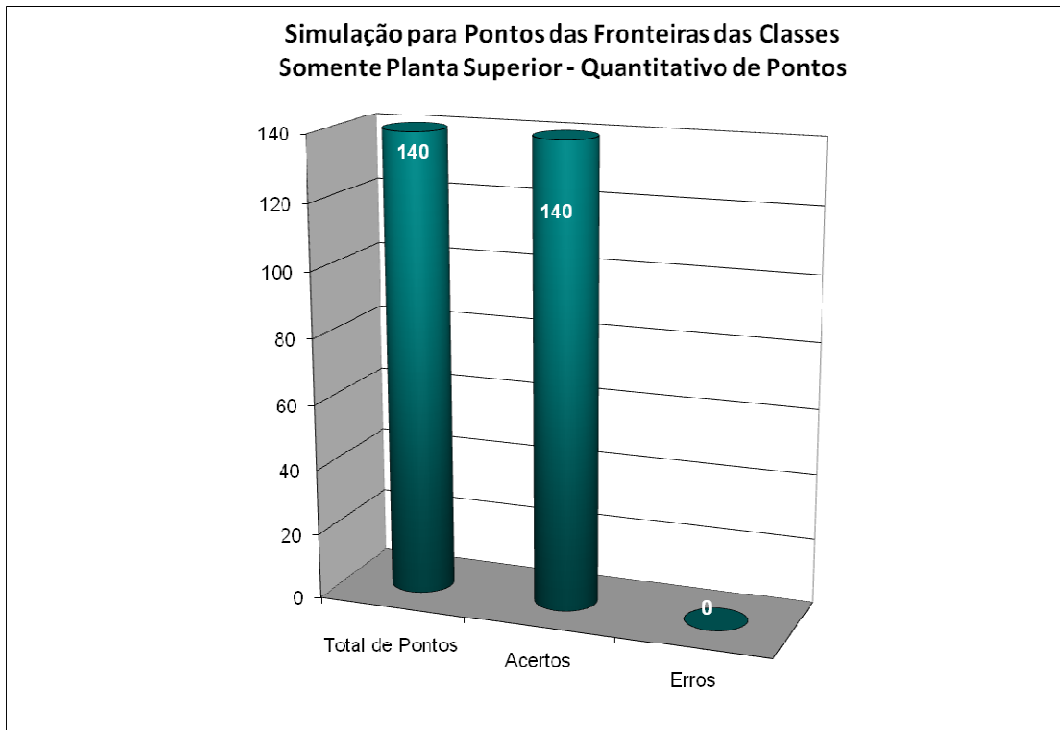


Figura 5.29 - Simulação para Pontos das Fronteiras das Classes - Somente Planta Superior - Quantitativo de Pontos – RNA Kohonen – 100% de Acerto

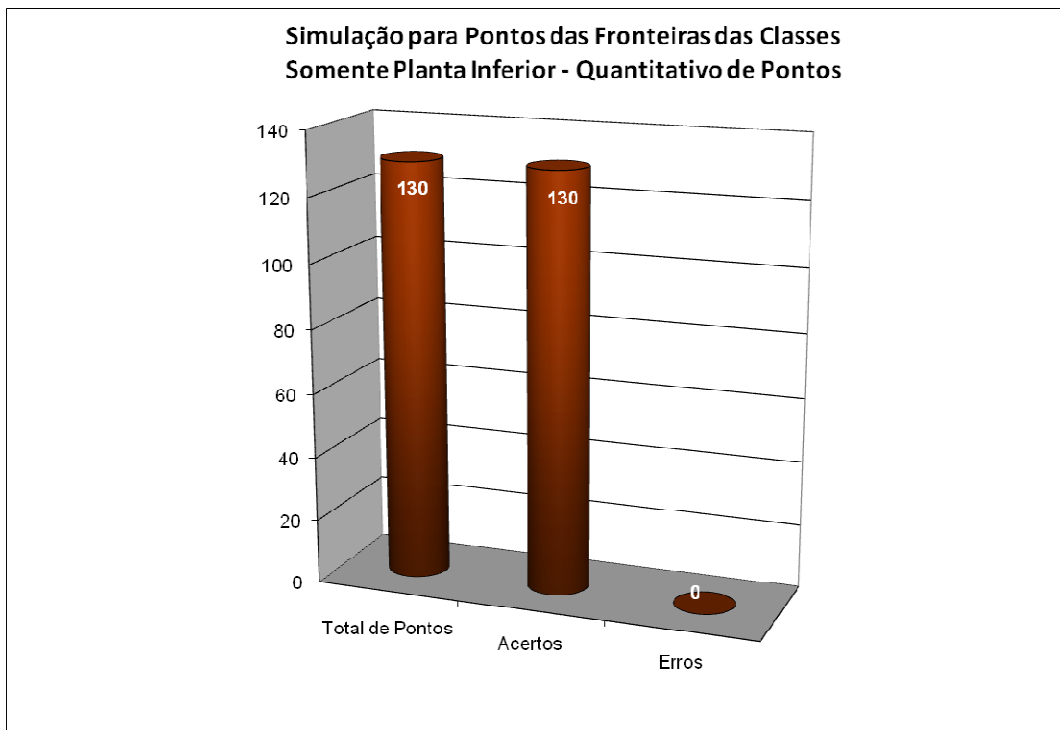


Figura 5.30 - Simulação para Pontos das Fronteiras das Classes - Somente Planta Inferior - Quantitativo de Pontos – RNA Kohonen – 100% de Acerto

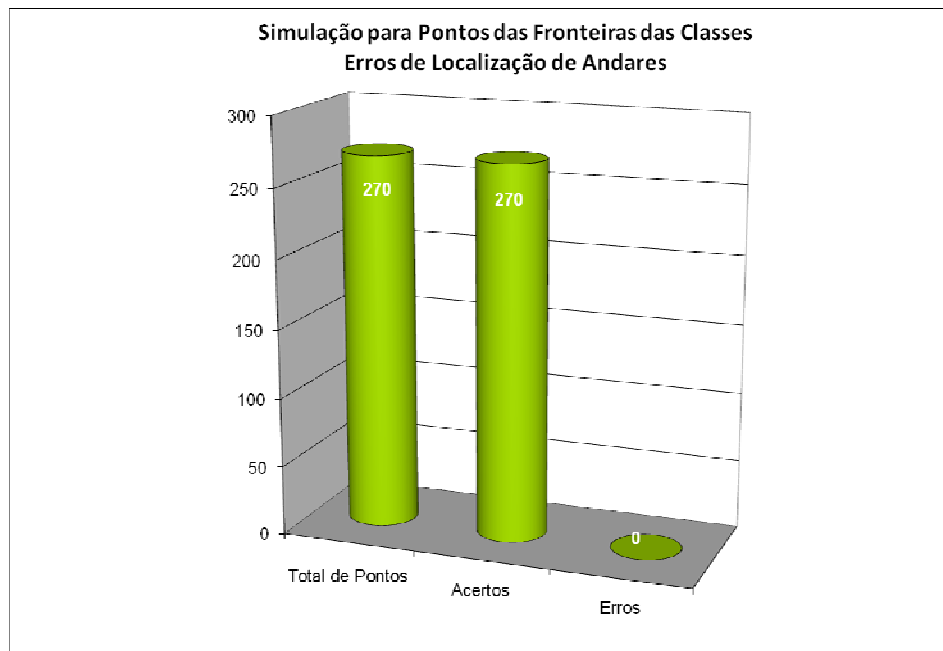


Figura 5.31 - Simulação para Pontos das Fronteiras das Classes - Planta Completa (Consolidada) - Quantitativo de Pontos – RNA Kohonen – 100% de Acerto

Quando tratamos dos problemas de pontos localizados em regiões fronteiriças, a rede comportou-se perfeitamente, não apresentando erros para os pontos simulados (Figuras 5.29 a 5.31). Não ocorreram erros de localização entre andares. Isso nos mostra que o treinamento da rede foi adequado, os parâmetros apresentados nessa fase são adequados para as medidas e estas, devidamente tratadas antes do processo de treino. Com isso, o mapa auto-organizável foi capaz de criar classes bastante definidas, e mesmo pontos localizados nas regiões entre uma classe e outra, não são capazes de prejudicar o desempenho da rede.

Entretanto, é importante destacar que a rede neural não é infalível. Ela não apresentou erros para o caso em análise por conta dos motivos expostos no parágrafo anterior. Apesar disso, caso o processo de simulação tivesse se iniciado antes mesmo da identificação dos melhores parâmetros para o treino (como exemplo), possivelmente os resultados não seriam semelhantes aos apresentados nesse estudo.



### 5.1.3.3 - Simulação contendo novos pontos adquiridos em novas medidas de campo aleatórias

Para testar a capacidade de generalização da rede neural, fomos novamente a campo para adquirir novas medidas de *RSSI* para compor um novo banco de testes para utilização nesta fase. O critério utilizado foi adquirir 50% do total da quantidade de medidas originais em cada andar em locais distintos. Com isso, adquirimos 59 novos pontos na planta superior e 51 novos pontos para a planta inferior. Os resultados encontram-se a seguir.

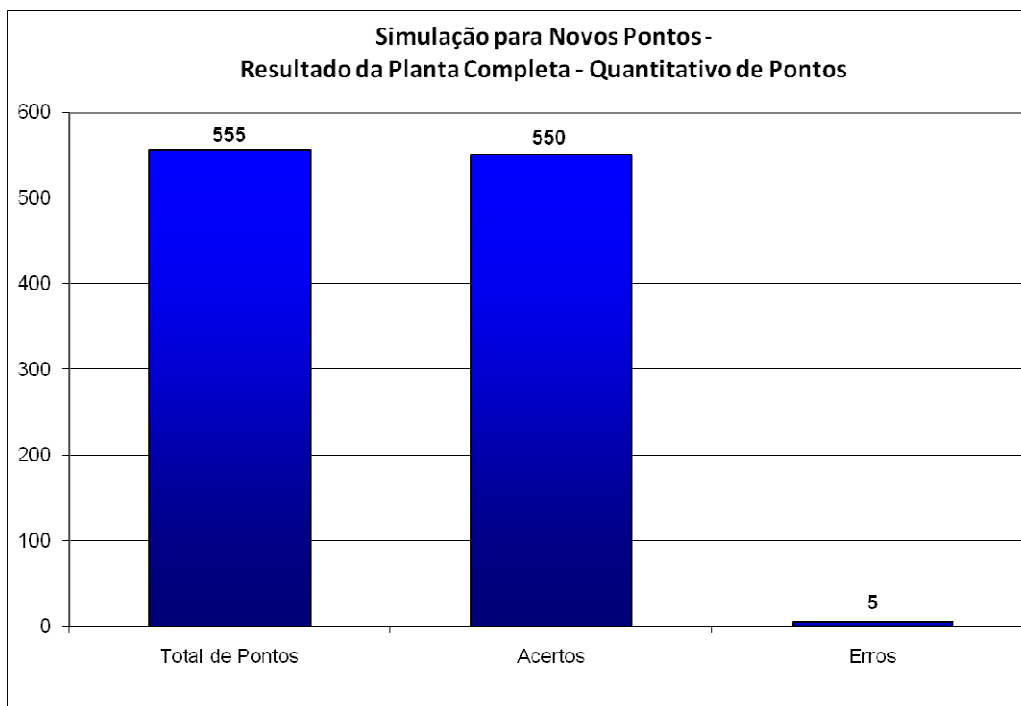


Figura 5.32 - Simulação para Novos Pontos - Resultado da Planta Completa – Total de Pontos – RNA Kohonen – 99,10% de Acerto e 0,9% de Erro

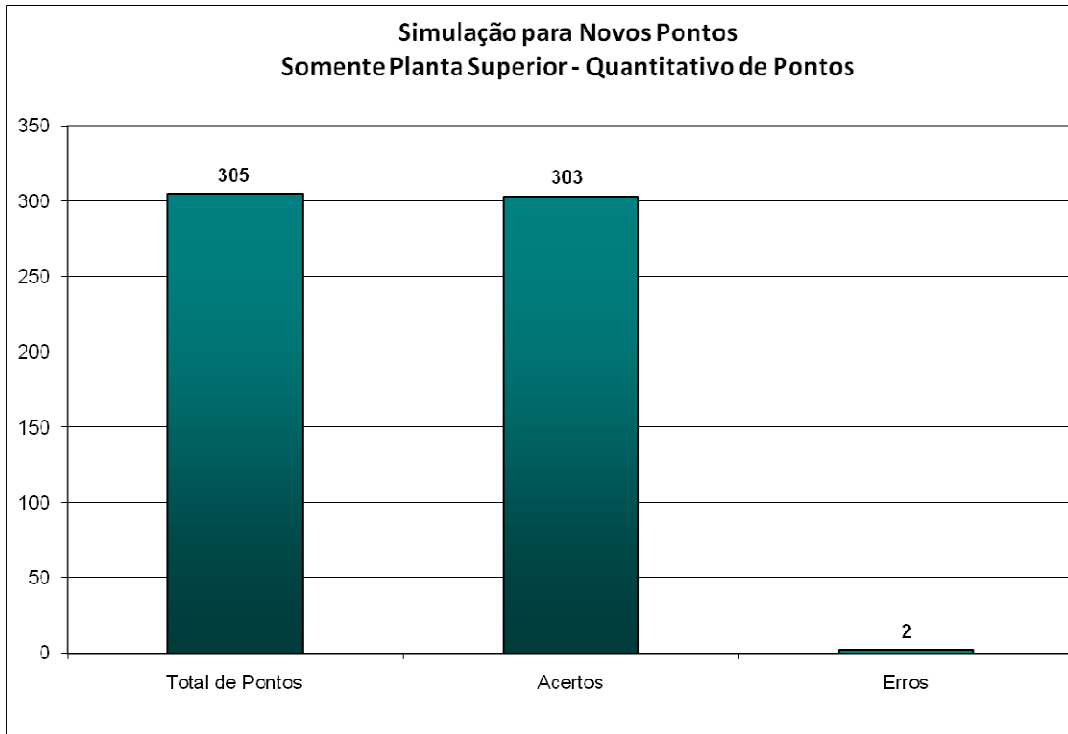


Figura 5.33 - Simulação para Novos Pontos – Somente Planta Superior – Total de Pontos – RNA Kohonen – 99,34% de Acerto e 0,66% de Erro

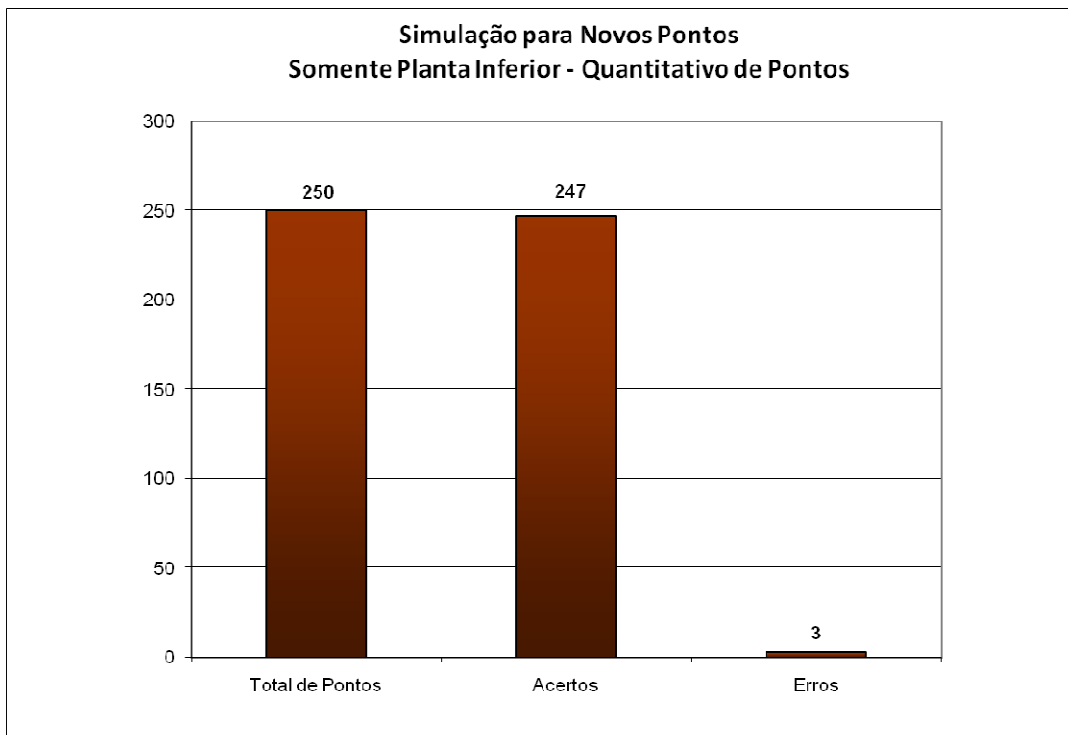


Figura 5.34 - Simulação para Novos Pontos – Somente Planta Inferior – Total de Pontos – RNA Kohonen – 98,80% de Acerto e 1,2% de Erro

Por fim, quando tratamos de pontos não presentes na simulação, a rede continua apresentando resultados bastante satisfatórios. Não ocorreram erros de localização entre andares. Do total de 555 novos pontos adquiridos experimentalmente, a rede errou em 0,91% deles, ou 5 pontos (Figuras 5.32 a 5.34). Quando analisamos a motivação desses erros, percebemos que grande parte deles foi influenciada pelo processo de medição. Essas medidas foram inseridas na rede neural com bastante degradação (ruído), e por isso, apesar do acerto na maioria delas, a rede errou em cinco delas.

A seção seguinte trata da aplicação da Rede Neural Multicamadas para o mesmo problema. Os resultados serão analisados isoladamente, e em seguida, comparados com a RNA de Kohonen.

## **5.2 - REDE NEURAL PERCEPTRON MULTICAMADAS (MLP)**

Como vimos na seção anterior, a rede neural de Kohonen teve erro máximo de 4% no problema da localização. Nessa etapa, o objetivo foi criar uma rede neural Perceptron multicamadas, com as mesmas condições de treinamento da rede anterior, e observar os resultados alcançados.

A bibliografia mais recente indica que para este tipo de rede neural, resultados da ordem de 10% a 15% de erro nas simulações são considerados normais [66]. Os experimentos a seguir irão revelar/constatar esta informação.

### **5.2.1 - Definição das Variáveis de Treinamento**

A definição de variáveis para a criação da rede neural MLP e posterior treinamento não é uma tarefa simples. Isso se deve ao fato de existir pouca ou quase nenhuma indicação de parâmetros recomendados para tentativas iniciais de criação da rede, como encontramos para as redes auto-organizáveis. Trata-se de um processo empírico. Por este motivo, foram necessários diversos treinamentos com pequenas alterações em parâmetros de entrada da rede para que pudessemos encontrar os parâmetros adequados do treinamento.

Da mesma forma que trabalhamos com a rede de Kohonen, dividimos o arquivo de entrada em Planta Superior, Inferior e Completa. Criamos três redes neurais distintas, o comportamento de cada uma foi verificado isoladamente, e logo após, extraídos a combinação de parâmetros adequados para criação e treino das RNAs.

A função de transferência empregada para o problema foi a **tangente hiperbólica**. Conforme literatura especializada [30], ela é bem mais rápida que a função logística, além disso, o problema em questão possui a característica de que a saída apresenta valores positivos e negativos que podem ser interessantes para a camada seguinte.

A função de treinamento selecionada foi a **Levenberg-Marquardt Backpropagation**. Essa função atualiza os pesos e estados de viés (*bias*) por meio da otimização de Levenberg-Marquardt. Ela é considerada a função mais rápida utilizada em simulações de redes neurais, altamente recomendada para tipos de treinamento onde o aprendizado é supervisionado e onde não há limitações acentuadas de memória de processamento [30].

Por fim, como o método de treinamento da rede é baseado na divisão das amostras em testes, validação e treino, um algoritmo que faça a divisão dessas amostras é importante. Essa divisão foi realizada por meio do algoritmo que trabalha com a **divisão dinâmica das amostras** [30].

Em seguida, as análises dos resultados dos treinamentos consideraram o **valor da média dos erros** em relação à **quantidade de neurônios**. Busca-se aqui, encontrar a quantidade ideal de neurônios nas camadas intermediárias de forma a obter o menor erro possível.

### **5.2.2 - Resultados dos Treinamentos**

O objetivo dessa etapa é encontrar a melhor combinação de parâmetros para cada rede neural criada. Analogamente ao treinamento da rede neural Kohonen, as amostras de

entrada foram divididas de acordo com as plantas do SG-11. Sendo assim, o treinamento é composto por 3 etapas: Planta Superior, Inferior e Completa.

Nesse treinamento, buscamos o menor erro possível – calculado por meio da função baseada na média dos erros ao quadrado – onde a quantidade de neurônios presentes nas camadas intermediárias da rede será a solução para esta fase.

Em todos os treinamentos, a quantidade de neurônios foi aumentada de 10 em 10 – até atingir o valor de 50 - e logo em seguida, de 50 em 50, para as plantas superiores e inferiores iniciando com 10 neurônios. Para a Planta Completa, as variações são da ordem de 50 em 50 e de 100 em 100 neurônios iniciando com 100 deles.

No que tange à divisão de amostras para o treinamento, a configuração que trouxe melhores resultados consta da tabela 5.3.

Tabela 5.3 - Parâmetros de divisão de amostras para treinamento para Rede MLP  
(Plantas Superior, Inferior e Completa)

Evento	Percentual de Amostras
Treinamento	75%
Validação	12,5%
Testes	12,5%

#### 5.2.2.1 - Treinamento com a Planta Superior – SG11

O procedimento de treinamento foi iniciado com a planta superior. As figuras 5.35 e 5.36 apresentam o resultado final do treinamento. A evolução dos demais treinamentos está documentada nas tabelas 5.4 a 5.6 da seção 5.2.2.4.

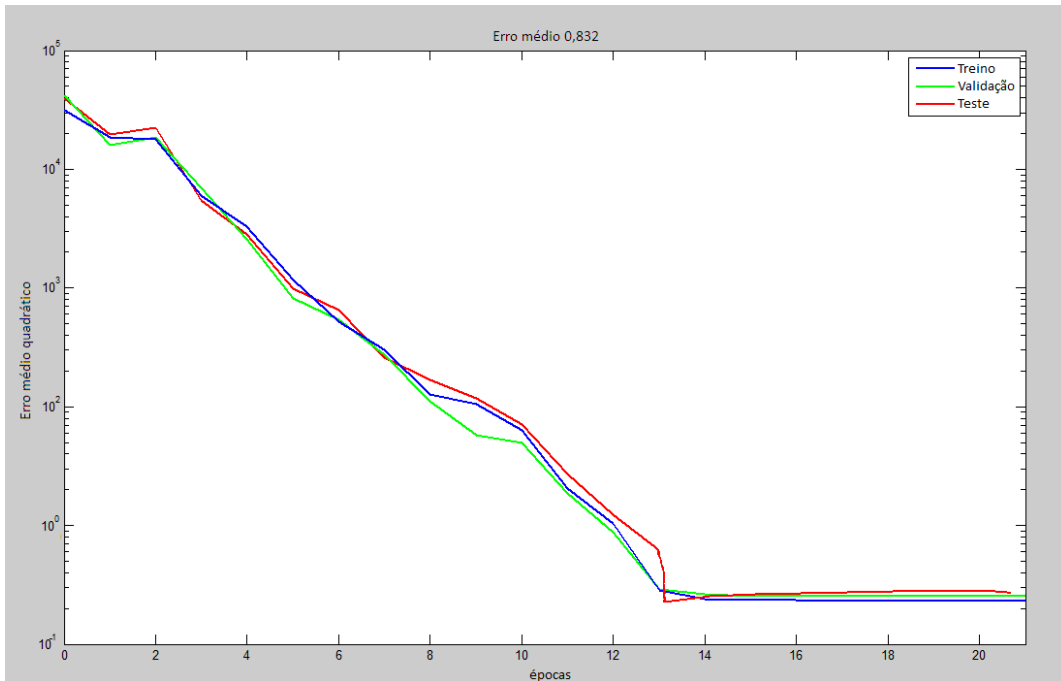


Figura 5.35 - Treinamento da Planta Superior com 150 neurônios nas camadas intermediárias (Melhor configuração) – Rede MLP

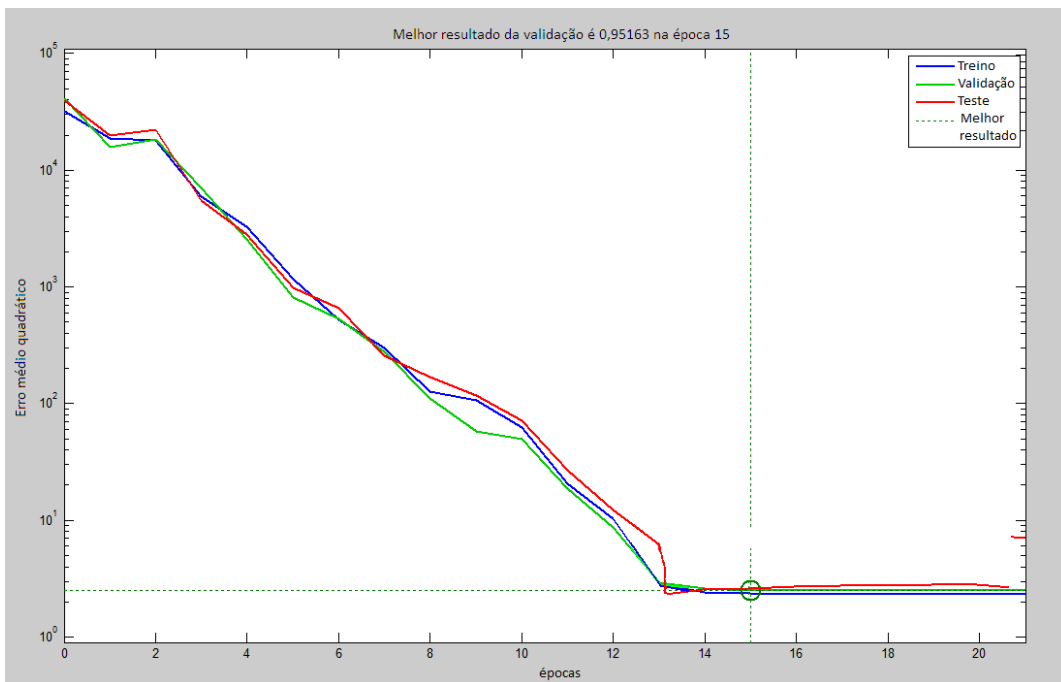


Figura 5.36 - Treinamento da Planta Superior com 150 neurônios nas camadas intermediárias (Melhor Resultado da Validação) – Rede MLP

### 5.2.2.2 - Treinamento com a Planta Inferior – SG11

Do mesmo modo, as figuras 5.37 e 5.38 representam o resultado final do treinamento com a planta inferior. A evolução dos demais treinamentos está documentada nas tabelas 5.4 a 5.6 da seção 5.2.2.4.

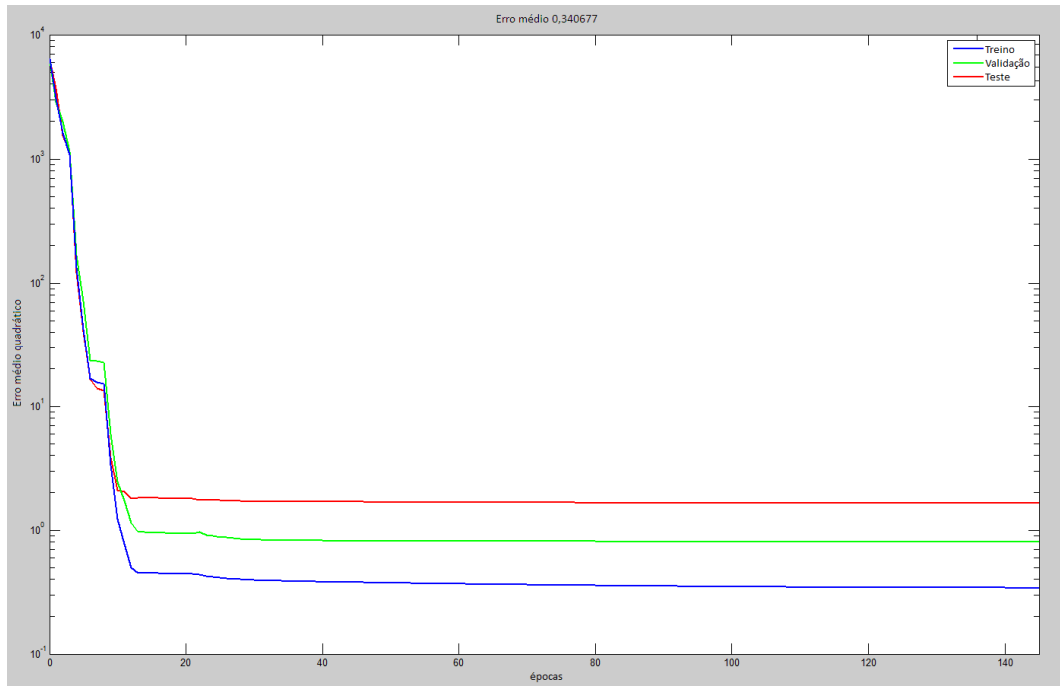


Figura 5.37 - Treinamento da Planta Inferior com 200 neurônios nas camadas intermediárias (Melhor configuração) – Rede MLP

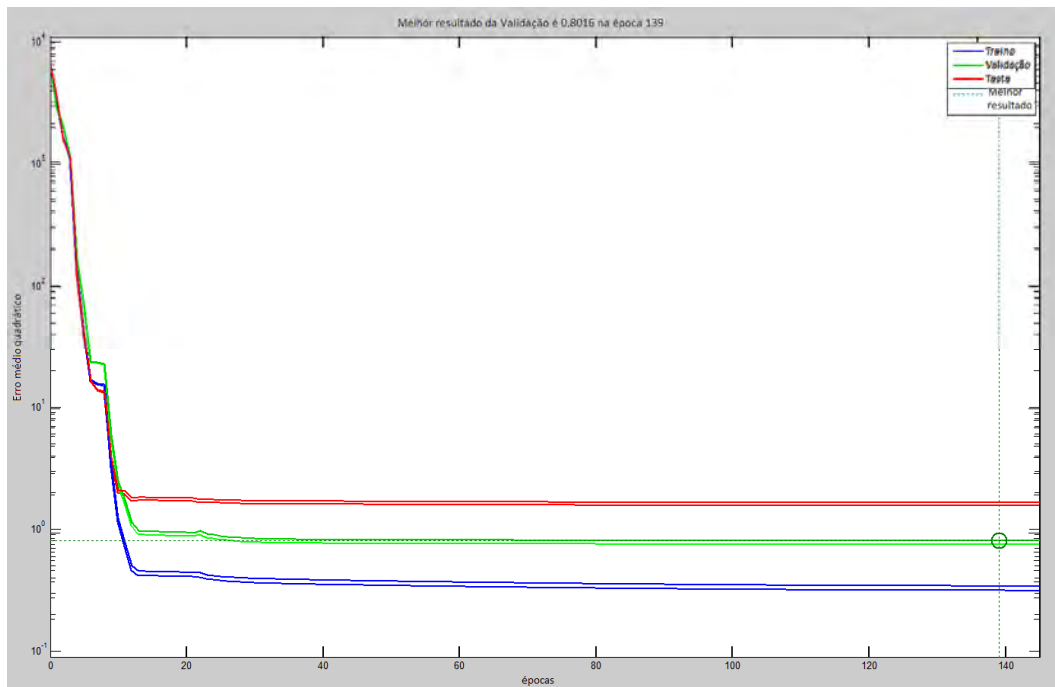


Figura 5.38 - Treinamento da Planta Inferior com 200 neurônios nas camadas intermediárias (Melhor Resultado da Validação) – Rede MLP

### 5.2.2.3 - Treinamento com a Planta Completa – SG11

Por fim, as figuras 5.39 e 5.40 representam o resultado final do treinamento com a planta completa. A evolução dos demais treinamentos está documentada nas tabelas 5.4 a 5.6 da seção “Considerações a respeito da fase de treinamento”.



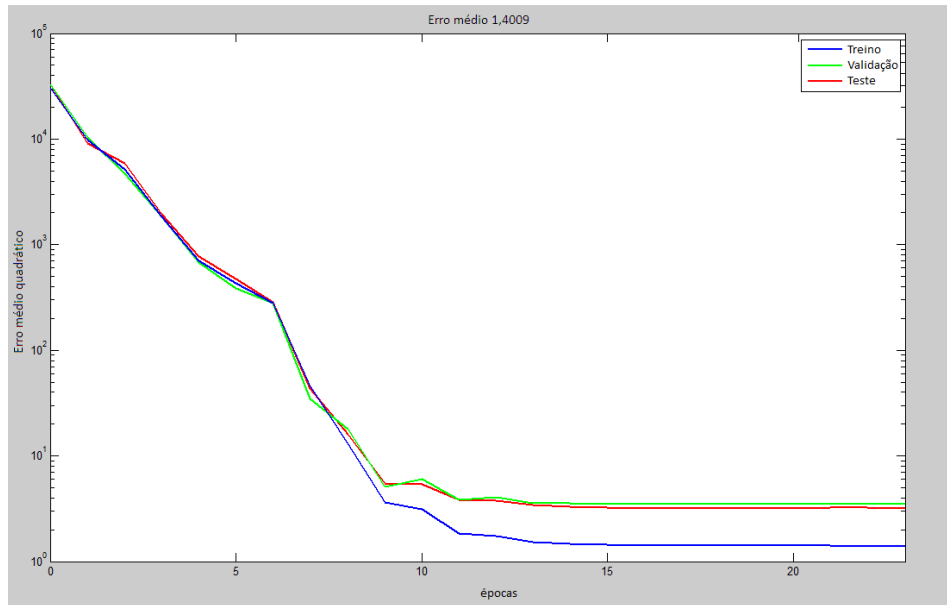


Figura 5.39 - Treinamento da Planta Completa com 250 neurônios nas camadas intermediárias (Melhor configuração) – Rede MLP

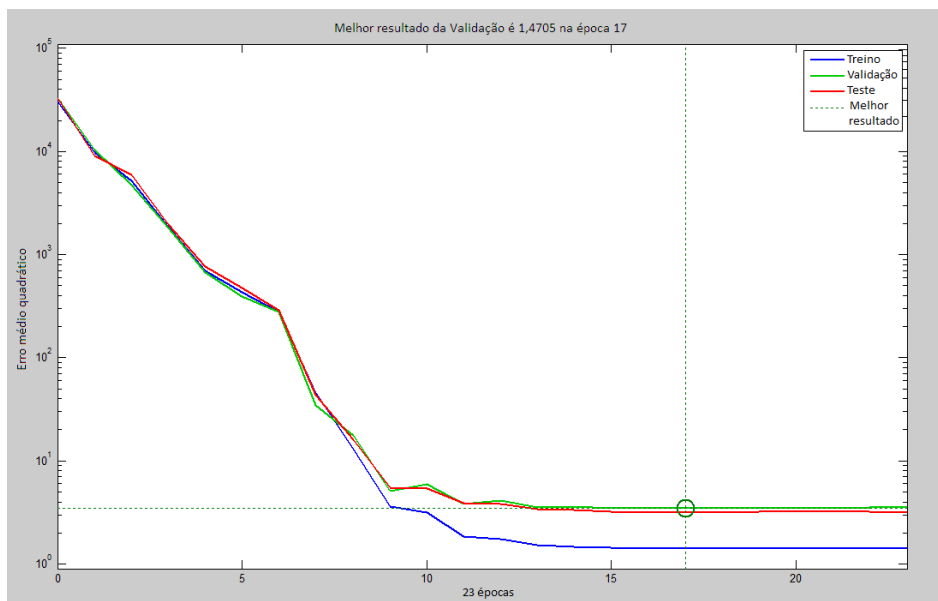


Figura 5.40 - Treinamento da Planta Completa com 250 neurônios nas camadas intermediárias (Melhor Resultado da Validação) – Rede MLP

#### 5.2.2.4 - Considerações a respeito da fase de Treinamento

Uma das primeiras diferenças encontradas na rede MLP em relação a Rede de Kohonen é o tempo de execução do algoritmo. De maneira geral, os treinamentos são mais

rápidos do que Kohonen, entretanto, os resultados alcançados em termos de precisão são inferiores, confirmando as informações da literatura.

As tabelas 5.4 a 5.6 consolidam todos os dados apresentados nas figuras passadas e indica a melhor configuração encontrada após todos os treinamentos realizados.

Observamos também que, nas etapas iniciais de treinamento, o simples aumento da quantidade de neurônios na rede não garante melhores resultados (Tabelas 5.4 a 5.6).

Tabela 5.4 - Resultados do Treinamento para Planta Superior – Rede Neural MLP

Resultados do Treinamento para Planta Superior - Rede Neural MLP					
Local	Iterações	MSE	Treino	Neurônios	Tempo de Execução [h:mins:s]
Planta Superior	40	5,25	5,934	10	00:12:02
	135	3,25	2,27	20	00:28:10
	86	2,89	1,59	30	00:22:08
	92	2,46	3,78	50	00:25:05
	53	0,9248	1,0679	100	00:19:08
	<b>21</b>	<b>0,832</b>	<b>0,95163</b>	<b>150</b>	<b>00:45:21</b>
	17	2,3	6,76	200	00:42:45
	27	2,48	2,51	250	00:51:10
	23	2,36	5,4747	300	00:51:29
	110	2,31	2,8503	350	00:59:13
	17	2,37	3,3	400	01:12:03
	145	2,46	2,6485	600	00:53:14
	17	2,5	1,249	700	00:57:22

Tabela 5.5 - Resultados do Treinamento para Planta Inferior – Rede Neural MLP

Resultados do Treinamento para Planta Inferior - Rede Neural MLP					
Local	Iterações	MSE	Treino	Neurônios	Tempo de Execução [h:min:s]
Planta Inferior	32	3,28	3,6717	10	00:21:02
	97	1,73	2,2374	20	00:35:04
	175	0,5937	0,3557	30	00:59:27
	87	0,9156	1,5938	50	00:30:17
	21	0,5	1,13	100	00:15:16
	196	0,3851	0,893	150	01:05:01
	<b>145</b>	<b>0,3406</b>	<b>0,8016</b>	<b>200</b>	<b>00:53:42</b>
	178	0,401	0,0402	250	01:25:46
	29	0,4605	1,5544	300	01:11:42
	67	0,4111	0,50216	350	01:15:18
	25	0,4893	1,0215	400	01:22:48
	11	0,423	1,0404	600	01:23:31
	43	0,406	0,95	700	01:28:26

Tabela 5.6 - Resultados do Treinamento para Planta Completa – Rede Neural MLP

Resultados do Treinamento para Planta Completa - Rede Neural MLP					
Local	Iterações	MSE	Treino	Neurônios	Tempo de Execução [h:min:s]
Planta Completa	237	1,5439	1,7382	100	01:33:27
	94	1,5407	2,3088	150	00:42:44
	23	1,4837	2,5077	200	00:51:22
	<b>23</b>	<b>1,4009</b>	<b>1,4705</b>	<b>250</b>	<b>01:01:30</b>
	21	1,5235	2,3624	300	00:59:38
	54	1,5283	3,2457	350	00:49:04
	16	1,5754	2,175	400	01:34:29
	22	1,5627	10,6088	600	01:42:15
	20	1,5061	1,6206	700	01:56:46

### 5.2.3 - Simulação

O processo de simulação para a rede neural MLP utilizou o mesmo arquivo de entrada empregado na rede neural Kohonen. Seguimos também a mesma abordagem do estudo anterior, em que a simulação considera a rede neural criada com todos os dados (Planta Completa) e dividimos a simulação em 3 etapas: Simulação com 75% dos pontos

utilizados no treinamento, Simulação com pontos em regiões de fronteira e Simulação contendo novos pontos. A etapa de simulação em regiões de fronteira sofreu pequenas alterações (inclusão de pontos), uma vez que agora essas regiões são basicamente as divisões entre os ambientes e não mais as divisões entre as classes de neurônios vencedores do mapa auto-organizável.

Por este motivo, nas análises da rede MLP, além de compararmos a precisão das redes quanto à localização das regiões, também foi verificada a precisão da rede MLP em metros. Essa última análise foi realizada a título ilustrativo e não poderá ser confrontada com os resultados da Rede de Kohonen.

Em relação à saída da rede neural, temos algumas diferenças. No mapa de Kohonen, a saída era basicamente a classe a que pertencia o ponto. Assim, sabíamos unicamente a região e o andar onde o ponto estava localizado. Para a Rede MLP, por conta das características de treinamento dessa rede neural, a camada de saída apresenta a coordenada do ponto e o respectivo andar, no formato  $(x,y,z)$ . A coordenada nos permite também localizar a região/ambiente em que o ponto está localizado. Na planta baixa, a coordenada  $(0,0,0)$  corresponde ao primeiro valor da esquerda para direita (em  $x$ ) e de cima para baixo (em  $y$ ) do andar térreo da planta (coordenada  $z$  do par de coordenadas). Com isso, a coordenada  $(0,0,1)$ , representa o primeiro valor dos pares de coordenadas da sobreloja do SG-11. Com isso, a saída da rede – que se trata de uma coordenada- foi confrontada com a planta baixa do ambiente para avaliar se a saída correspondia ao local indicado na planta.

Para todas as simulações, gráficos que avaliam os resultados de cada etapa da simulação podem contemplar erros de localização de medidas entre andares (caso exista). Do contrário, considere que o quantitativo de pontos incorretos e/ou percentual de erro se refere unicamente a erros de localização ocorridos no mesmo piso em análise.

#### 5.2.3.1 - Simulação contendo pontos utilizados no treinamento

Neste primeiro momento, a rede MLP trabalhou com 75% das 4.320 medidas totais, massa de dados idêntica àquela utilizada para a simulação da rede de Kohonen. Os

resultados encontrados serão apresentados em gráficos (Figuras 5.41 a 5.43) contendo a análise quanto ao erro x acerto da localização por região do mapa, e logo em seguida, uma tabela contendo a análise da precisão da rede em metros para cada tipo de simulação.

Reforçamos que esta última análise consta no trabalho para fins ilustrativos, não podendo servir para comparação com a Rede Neural de Kohonen.

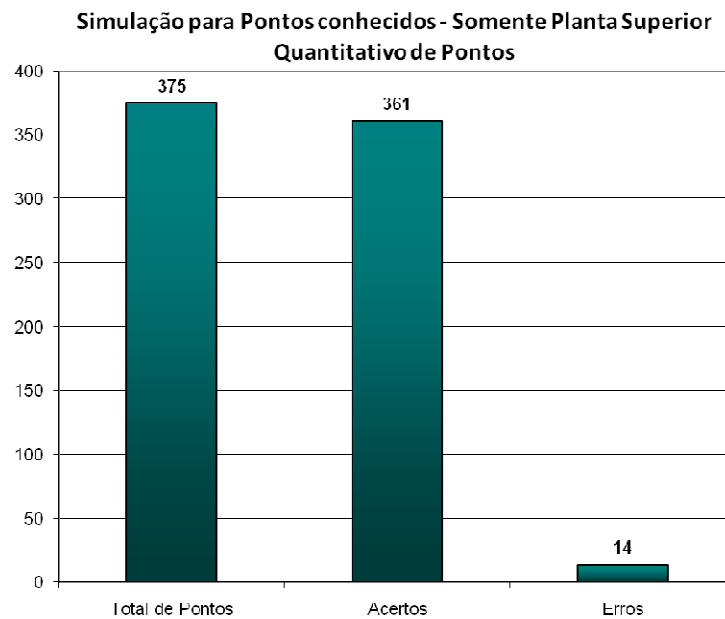


Figura 5.41 - Resultados alcançados para a Planta Superior – Quantitativos de Pontos – Rede MLP – 96,27% de Acerto e 3,73% de Erro

Tabela 5.7 – Erro médio em metros para Pontos Conhecidos - Planta Superior – Rede MLP

Local	Erro médio (metros) – Pontos Conhecidos	
<i>Planta Superior</i>	<b>Eixo X</b>	<b>Eixo Y</b>
	1,41	1,10

A tabela acima ilustra a média dos erros da rede em metros nos dois eixos. No que tange aos resultados quanto a localização de ambientes, a rede respondeu com 96,27 % de precisão e em relação a precisão em metros, tivemos um erro médio de 1,41 m em X e 1,10 m em Y.

Os gráficos abaixo ilustram o mesmo tipo de simulação (75% dos pontos do treinamento) para a RNA que responde pela **Planta Inferior**:

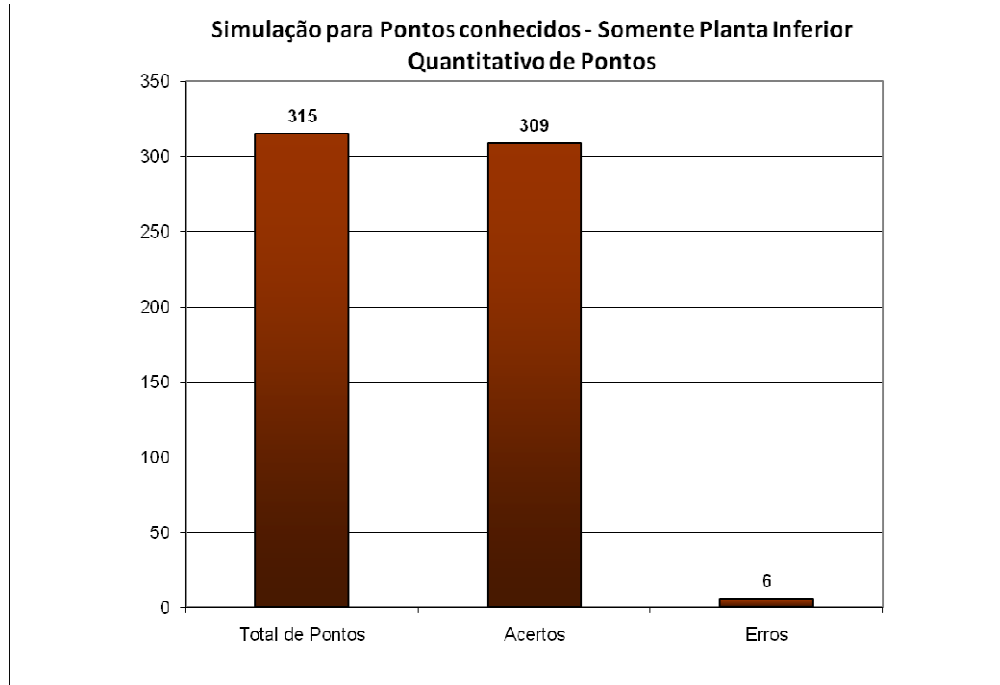


Figura 5.42 - Resultados alcançados para a Planta Inferior – Quantitativos de Pontos – Rede MLP – 98,10% de Acerto e 1,9% de Erro

Tabela 5.8 - Erro médio em metros para Pontos Conhecidos - Planta Inferior – Rede MLP

Local	Erro médio (metros) – Pontos Conhecidos	
	Eixo X	Eixo Y
<i>Planta Inferior</i>	0,39	0,46

Para a tarefa de localização de ambientes, a rede referente à **planta inferior** respondeu com 98,10 % de precisão e em relação à precisão em metros, tivemos um erro ligeiramente melhor para este caso: 0,39 m para o eixo X e 0,46 m. Não ocorreram erros de localização entre andares.

Em seguida, os resultados finais para essa etapa contemplando a **Planta Completa**:

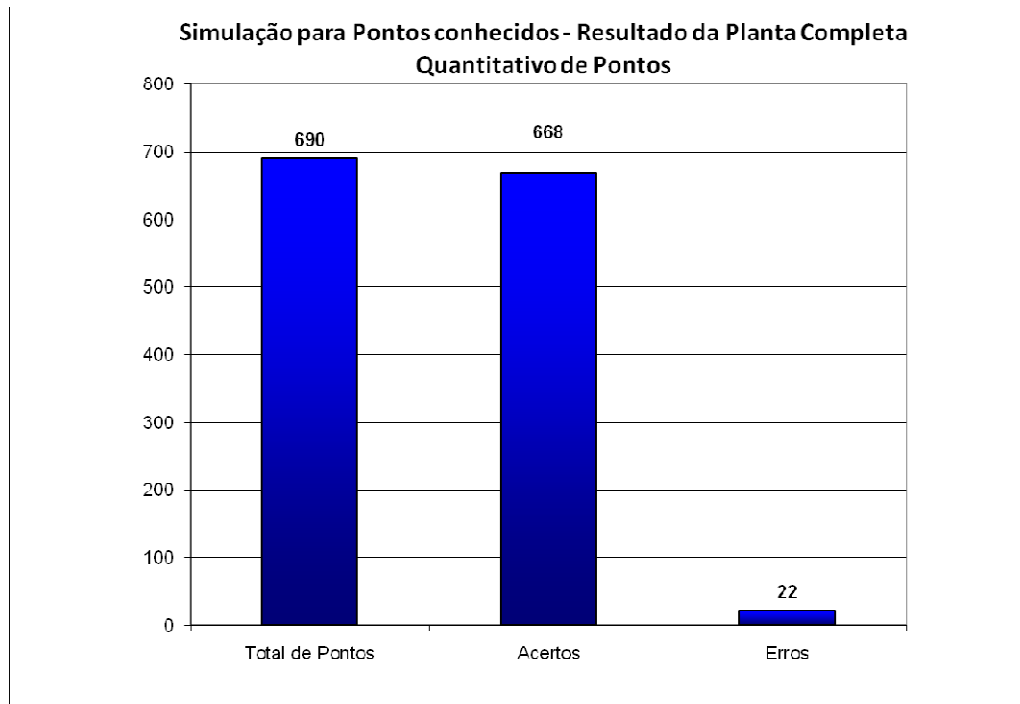


Figura 5.43 - Resultados alcançados para a Planta Completa – Quantitativos de Pontos – Rede MLP – 96,81% de Acerto e 3,19% de Erro

Tabela 5.9 - Erro médio em metros para Pontos Conhecidos - Planta Completa – Rede MLP

Local	Erro médio (metros) – Pontos Conhecidos	
	Eixo X	Eixo Y
<i>Planta Completa</i>	0,93	0,80

Nas simulações para pontos conhecidos, por mais que encontrássemos erros da ordem de 4 m, nas direções X e Y, ainda assim, as regiões afetadas eram aquelas localizadas em regiões de fronteiras de ambientes. Pontos que não pertenciam às fronteiras, mesmo com o erro da rede neural, ainda eram localizados dentro do ambiente correto. Por este motivo, o total de 3,29% de erro apresentado nessa simulação consiste em erros nas fronteiras.

### 5.2.3.2 - Simulação contendo pontos localizados em regiões de fronteira

Nessa etapa de simulação, o principal objetivo é a análise de possíveis de erros de localização em pontos adjacentes às paredes que dividem ambientes. Os dados de entrada sofreram um acréscimo em relação ao treinamento dispensado na rede de Kohonen.

Enquanto a rede de Kohonen classificou, em sua melhor configuração, dois ou mais locais dentro de uma mesma classe (por apresentar características semelhantes), a rede MLP trabalhará considerando a divisão dos ambientes dentro da planta baixa, onde os pontos adjacentes as paredes compõem as regiões de fronteira.

Com isso, o arquivo de entrada aumentou, contando agora com 88 pontos ou 440 medidas.

As figuras a seguir ilustram a localização desses pontos para as duas plantas e logo em seguida, os resultados com essa configuração são explorados.



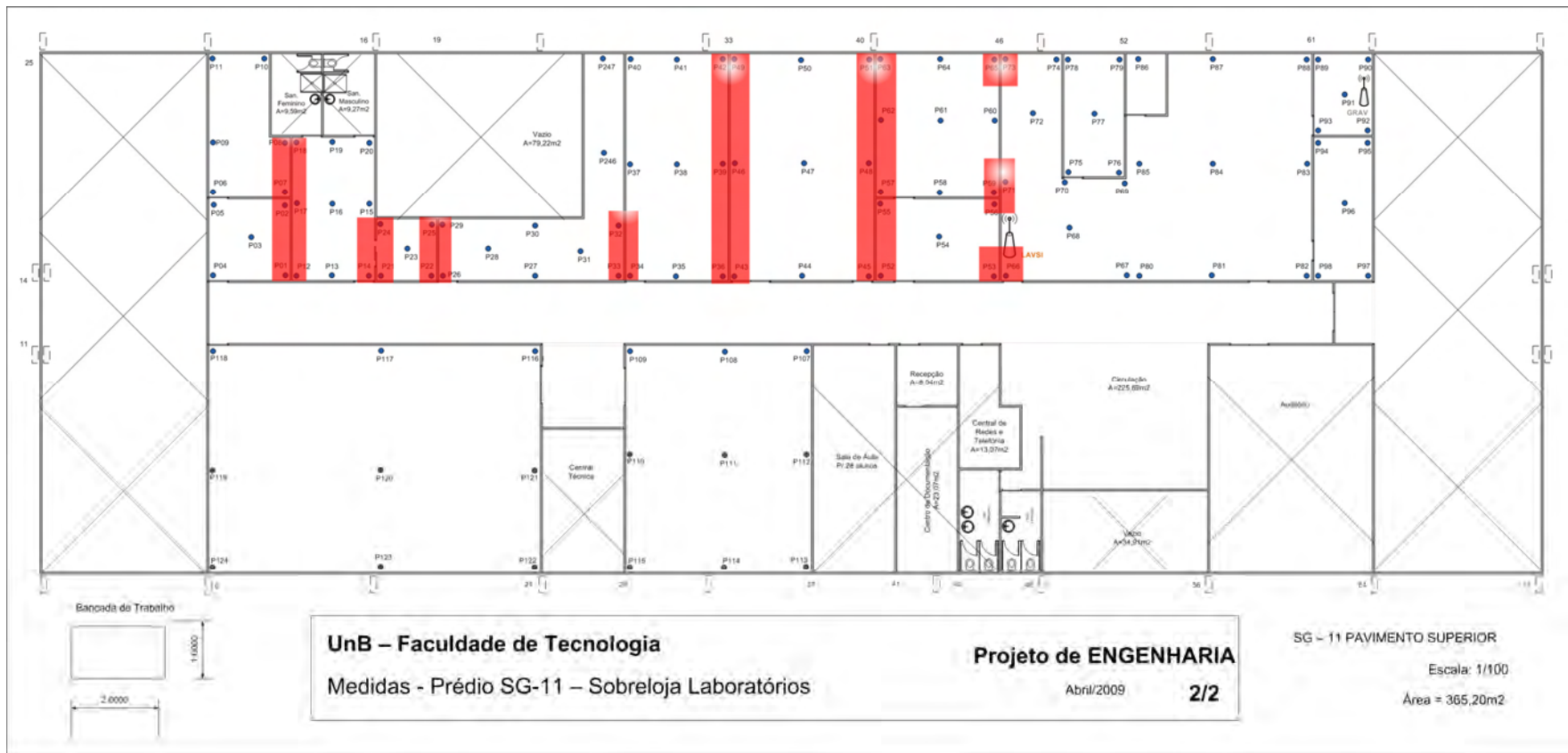


Figura 5.44 - Localização dos novos Pontos para as regiões de fronteira – Planta Superior – Rede MLP

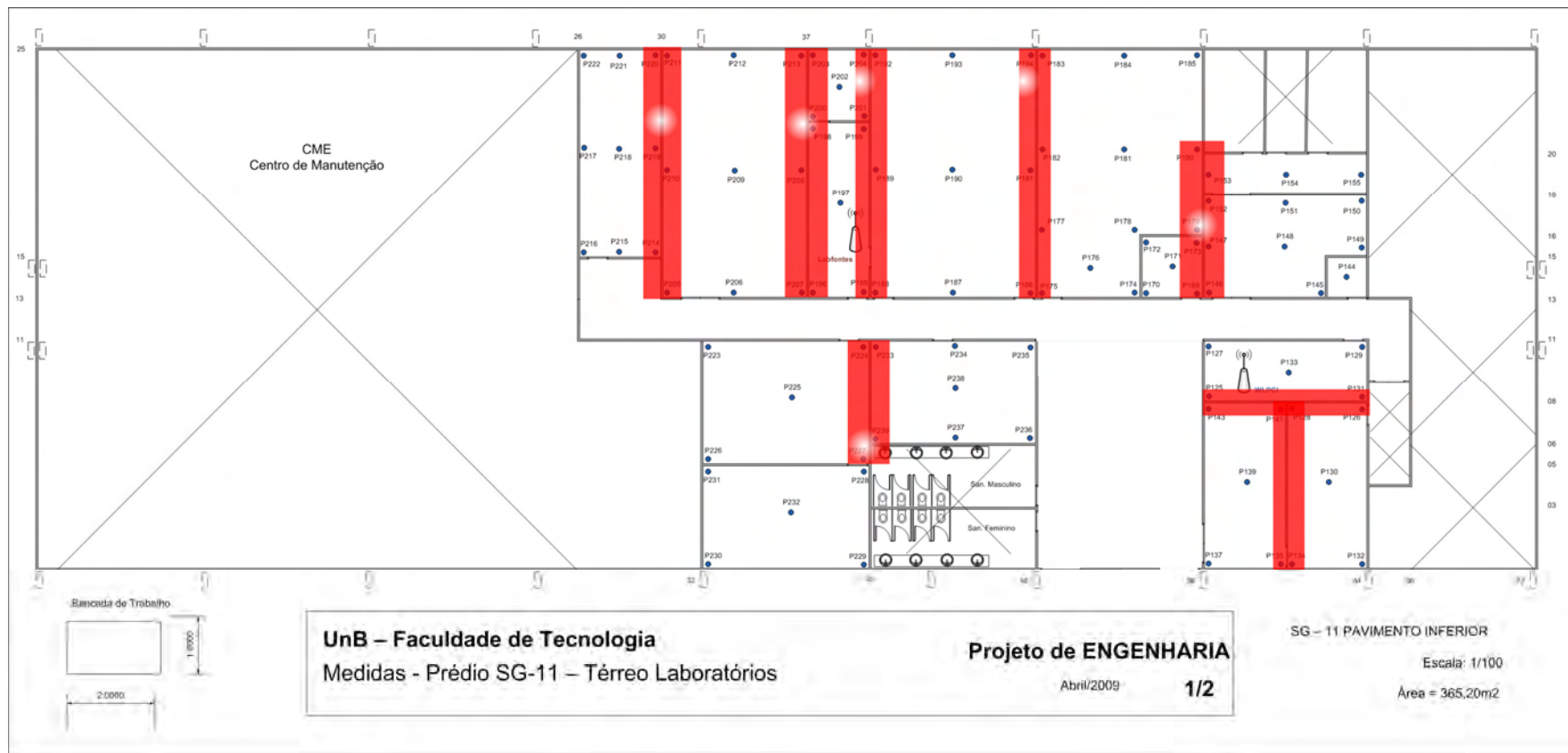


Figura 5.45 - Localização dos novos Pontos para as regiões de fronteira – Planta Inferior – Rede MLP

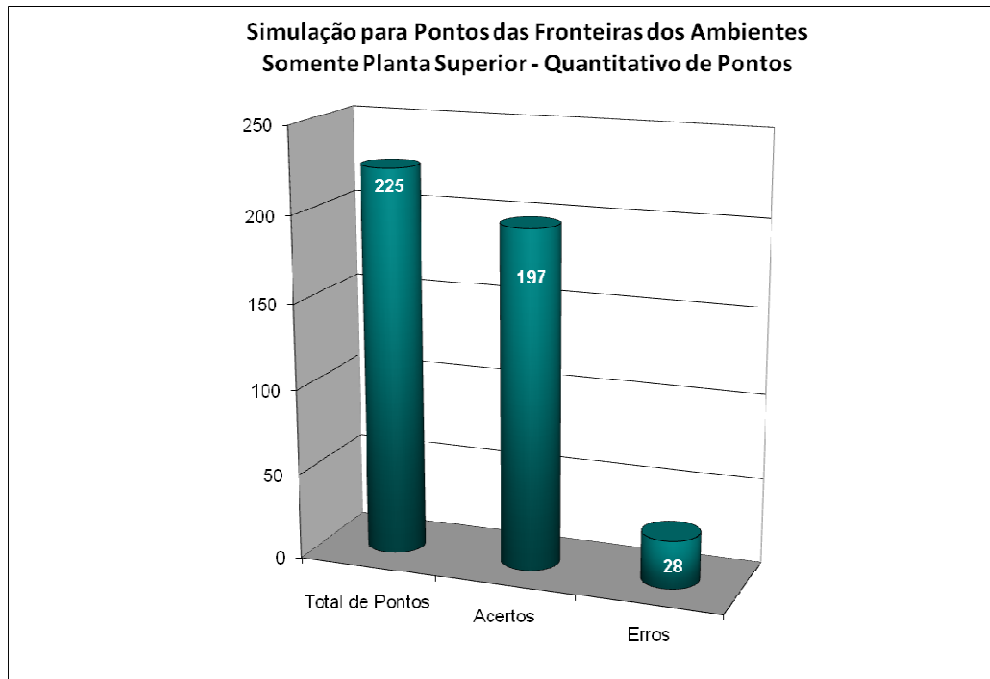


Figura 5.46 - Resultados alcançados para regiões de fronteira - Planta Superior – Quantitativo de Pontos – Rede MLP – 87,56% de Acerto e 12,44% de Erro

Tabela 5.10 - Erro médio em metros para Regiões de Fronteira - Planta Superior – Rede MLP

Local	Erro médio (metros) – Regiões de Fronteira	
	<i>Planta Superior</i>	<b>Eixo X</b>
	1,10	1,28

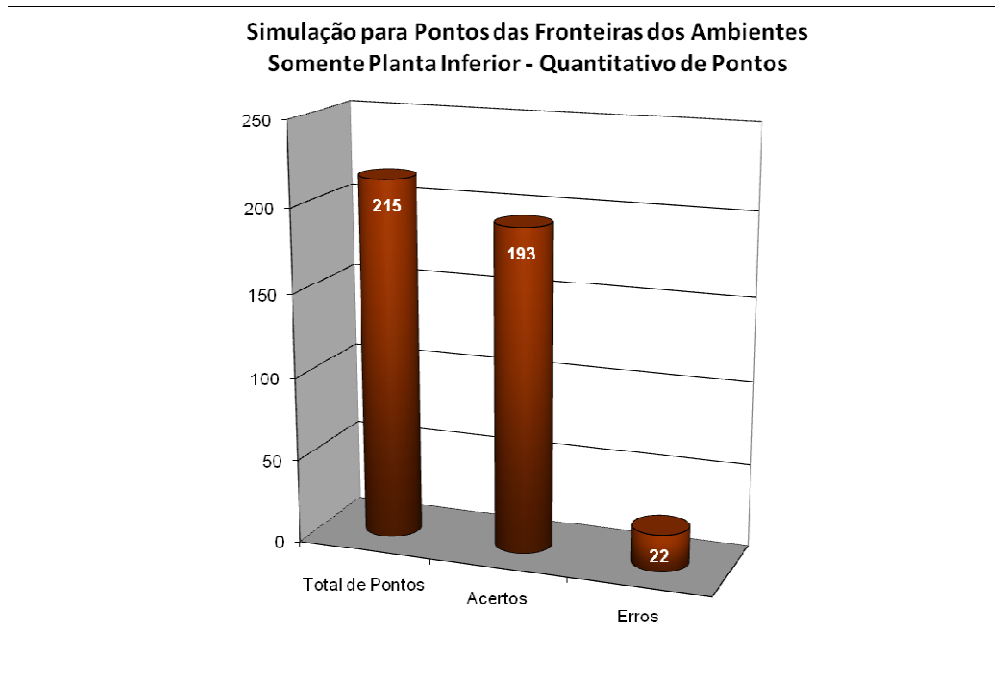


Figura 5.47 - Resultados alcançados para regiões de fronteira - Planta Inferior – Quantitativo de Pontos – Rede MLP – 89,77% de Acerto e 10,23% de Erro

Tabela 5.11 - Erro médio em metros para Regiões de Fronteira - Planta Inferior – Rede MLP

Local	Erro médio (metros) – Regiões de Fronteira	
	Eixo X	Eixo Y
<i>Planta Inferior</i>	0,33	0,42

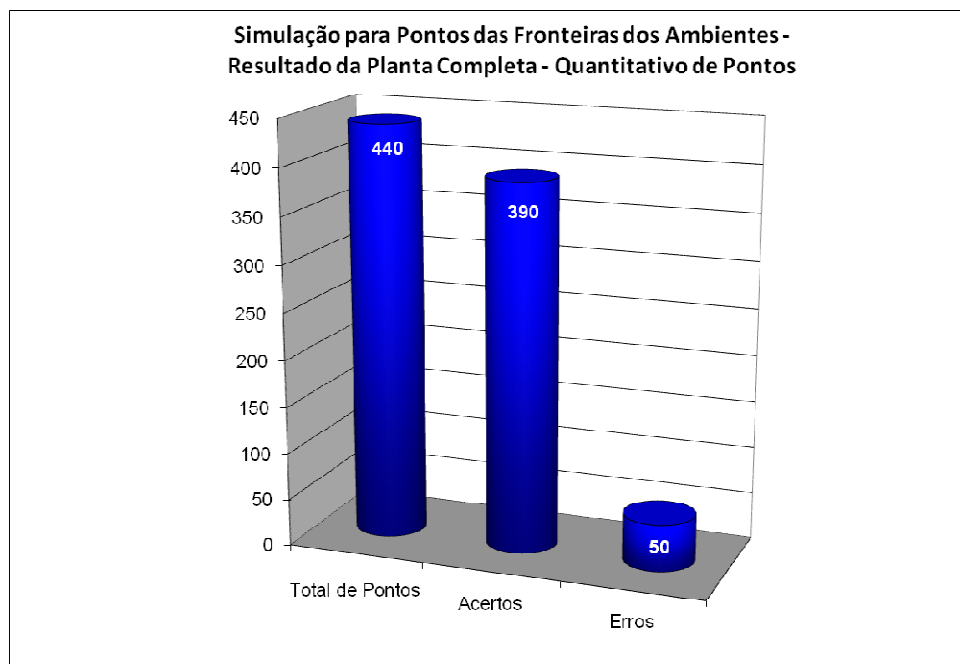


Figura 5.48 - Resultados alcançados para regiões de fronteira - Planta Completa – Quantitativo de Pontos – Rede MLP – 88,64% de Acerto e 11,36% de Erro

Tabela 5.12 - Erro médio em metros para Regiões de Fronteira - Planta Completa – Rede MLP

Local	Erro médio (metros) – Regiões de Fronteira	
	Eixo X	Eixo Y
<i>Planta Completa</i>	0,73	0,86

A Rede MLP não apresentou desempenho semelhante ou superior àquele apresentado nas simulações contendo pontos conhecidos. Nessa situação, encontramos erros na casa de 12,82%, representando 50 medidas com localização equivocada do total de 440. Os erros estão distribuídos praticamente em porções semelhantes, 28 para a sobreloja e 22 para o térreo.

A análise particular de cada ponto comprometido nos levou a duas conclusões. A primeira é relativa às 28 medidas da planta superior. A maior parte desses pontos está localizada em regiões que a Rede de Kohonen agrupou em uma só classe. A rede MLP tentando informar a coordenada geográfica do ponto, mesmo com uma boa precisão em metros, informou a posição deslocada tanto para X, quanto para Y, de forma que o ponto ficasse localizado fora do ambiente original, considerado erro para essas simulações. Para as 20 medidas restantes, apesar da quantidade menor de pontos e

ambientes mapeados, a rede não foi capaz de trabalhar com as medidas ruidosas de APs localizados na sobreloja (que compõem a entrada da rede). Com isso, a precisão para estes pontos, é ligeiramente prejudicada, e a rede erra por questões de metros. Nesse caso, é possível melhorar a precisão da rede com mais treinamentos e possivelmente novas medidas de campo para estes pontos.

#### 5.2.3.3 - Simulação contendo novos pontos adquiridos em novas medidas de campo aleatórias

Como última etapa para as simulações, a rede MLP utilizou medidas idênticas aquelas utilizadas na rede Kohonen para avaliação das novas medidas de campo. Nesse ensaio, o objetivo é verificar a capacidade de generalização da rede neural, possibilitando assim, identificar a necessidade de se utilizar mais ou menos dados de entrada para o treinamento da rede e avaliar o funcionamento (em termos de precisão) da rede em condições reais de uso.

De maneira geral, uma rede neural adequada para o problema e bem treinada é capaz de utilizar o conhecimento adquirido em treinamentos anteriores e responder, de forma bastante eficaz em simulações de pontos não conhecidos. Os gráficos a seguir fornecem insumos para a avaliação do desempenho desse tipo de rede ao problema proposto.

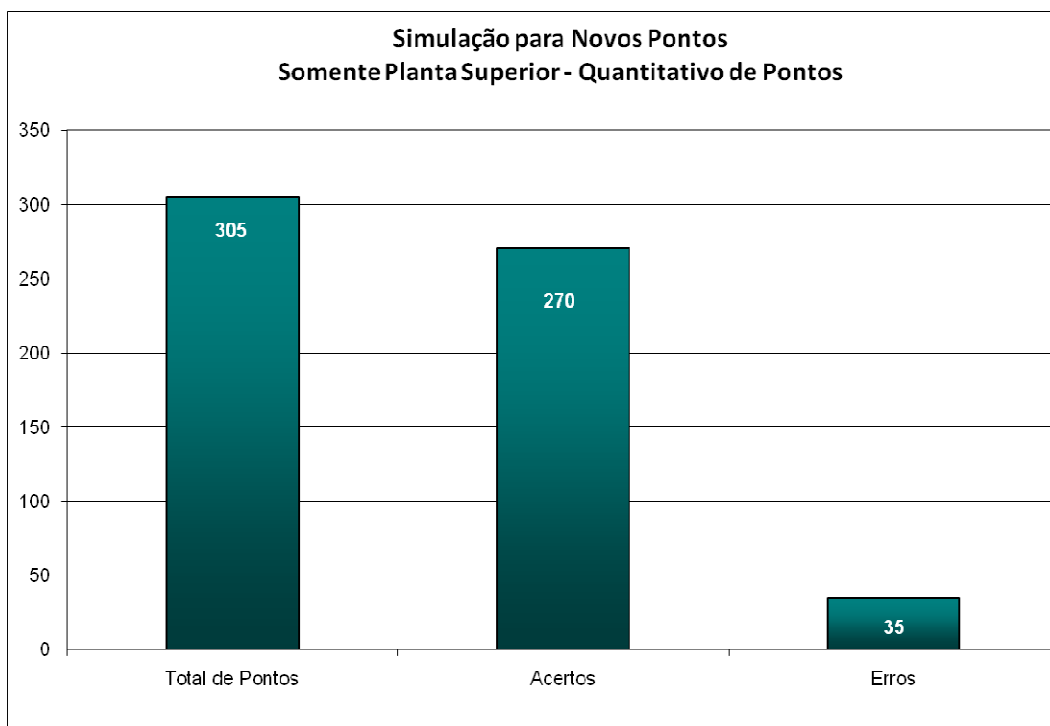


Figura 5.49 - Resultados alcançados para novos pontos - Planta Superior – Quantitativo de Pontos – Rede MLP – 88,52% de Acerto e 11,48% de Erro

Tabela 5.13 - Erro médio em metros para Novos Pontos - Planta Superior – Rede MLP

Local	Erro médio (metros) – Novos Pontos	
	Eixo X	Eixo Y
<i>Planta Superior</i>	1,23	1,16

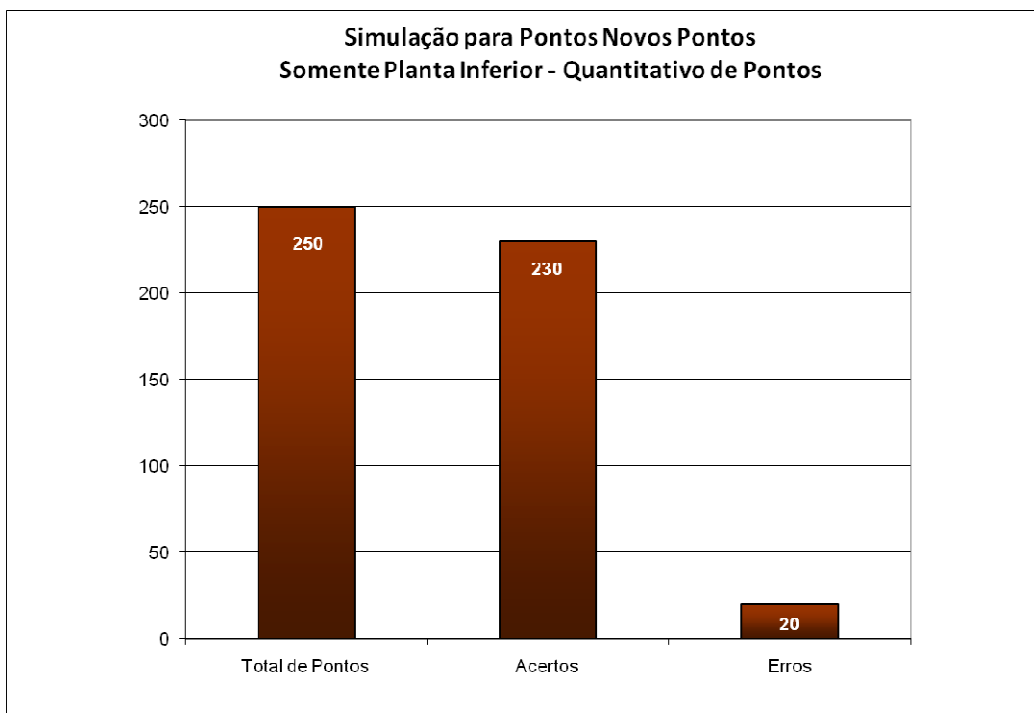


Figura 5.50 - Resultados alcançados para novos pontos - Planta Inferior – Quantitativo de Pontos – Rede MLP – 92% de Acerto e 8 % de Erro

Tabela 5.14 - Erro médio em metros para Novos Pontos - Planta Inferior – Rede MLP

Local	Erro médio (metros) – Novos Pontos	
	<i>Planta Inferior</i>	<b>Eixo X</b>
	0,88	0,81



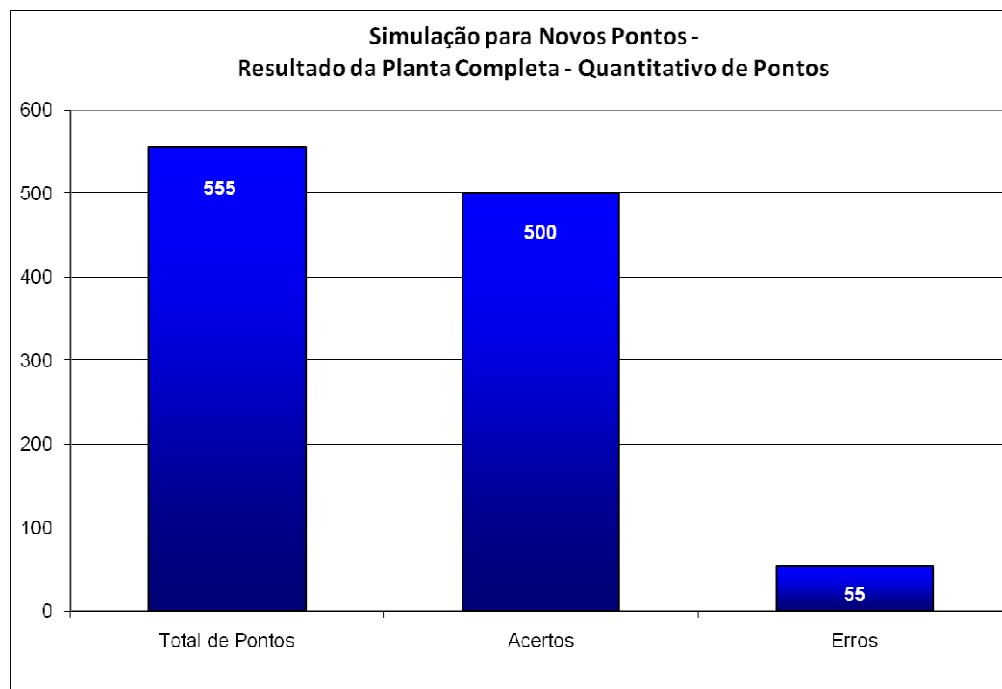


Figura 5.51 - Resultados alcançados para novos pontos - Planta Completa – Quantitativo de Pontos – Rede MLP – 90,09% de Acerto e 9,91% de Erro

Tabela 5.15 - Erro médio em metros para Novos Pontos - Planta Completa – Rede MLP

Local	Erro médio (metros) – Novos Pontos	
	Eixo X	Eixo Y
<i>Planta Completa</i>	1,07	0,99

A Rede MLP apresentou bons resultados para a situação apresentada. No conjunto, do total de 555 medidas representando 111 pontos, observamos um erro total de 55 medidas ou 11,00%.

Uma análise mais detalhada desses erros identificou que todos eles pertenciam a pontos próximos às paredes e em locais que dividem um ambiente de outro. Outra observação importante quanto aos erros é a distribuição deles entre os dois andares.

Para esta simulação (Novos Pontos), os erros se concentraram em maior quantidade na sobreloja (35 localizações incorretas) do que no andar térreo (20 localizações incorretas). Esse resultado era esperado desde o instante que analisamos os possíveis agrupamentos que a Rede de Kohonen faria com as características dos sinais em cada andar. O andar superior possuiu diversos intervalos que se sobrepõem, e os APs presentes no andar térreo causam certo ruído para a rede, o que não acontece com a

mesma intensidade para as medidas do andar inferior. Com isso, apesar do treinamento exaustivo realizado e a precisão média encontrada, ainda assim, a rede falhou em alguns pontos.

As duas imagens a seguir exibem a localização original dos pontos que causaram os erros descritos e a localização prevista pela rede. Observe que todos os pontos estão em regiões de fronteira e o resultado da simulação mostra estes mesmos pontos em regiões adjacentes às originais.

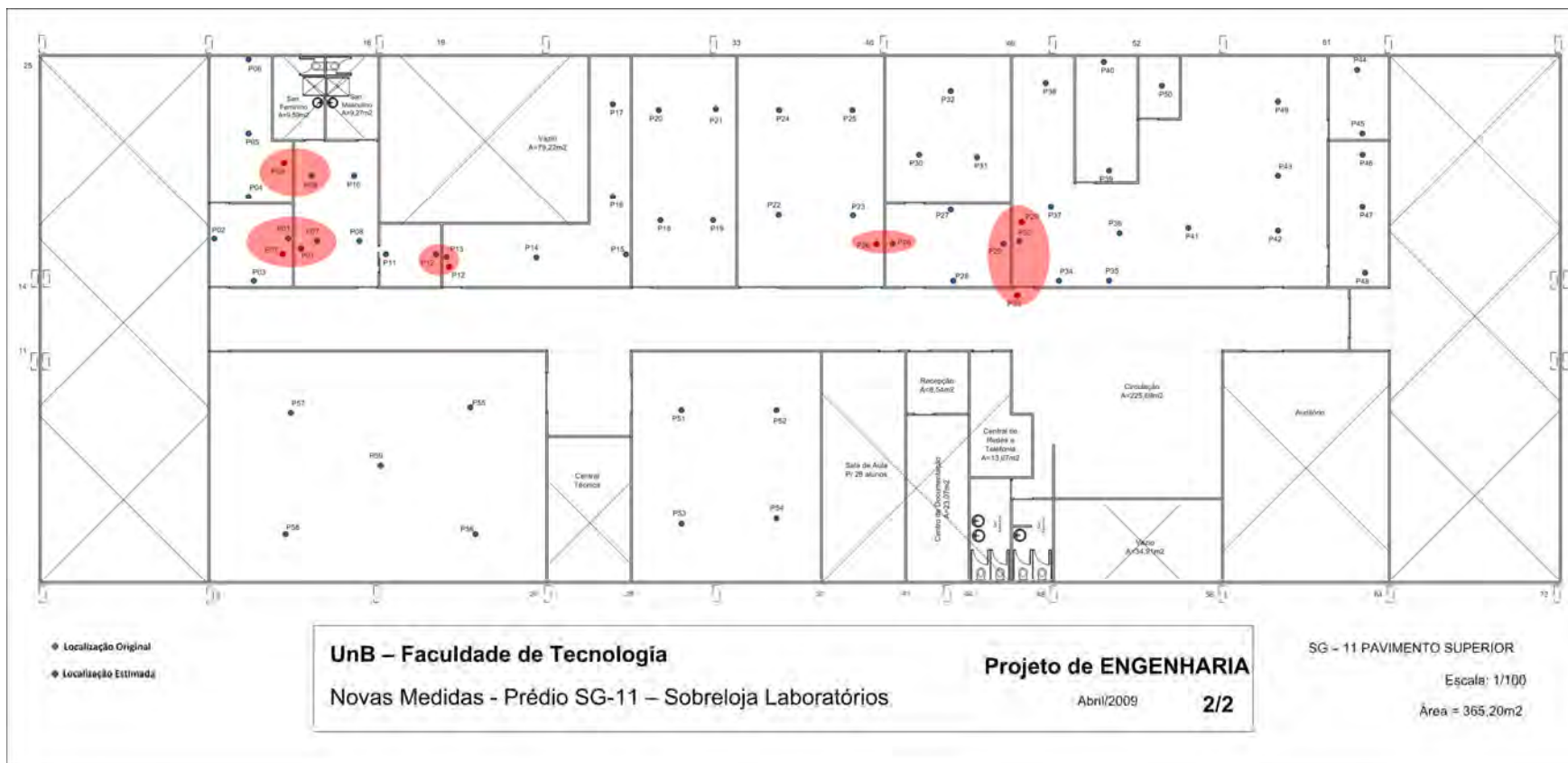


Figura 5.52 - Localização Original VS Localização Estimada pela RNA MLP – Planta Superior – 35 Medidas – Rede MLP

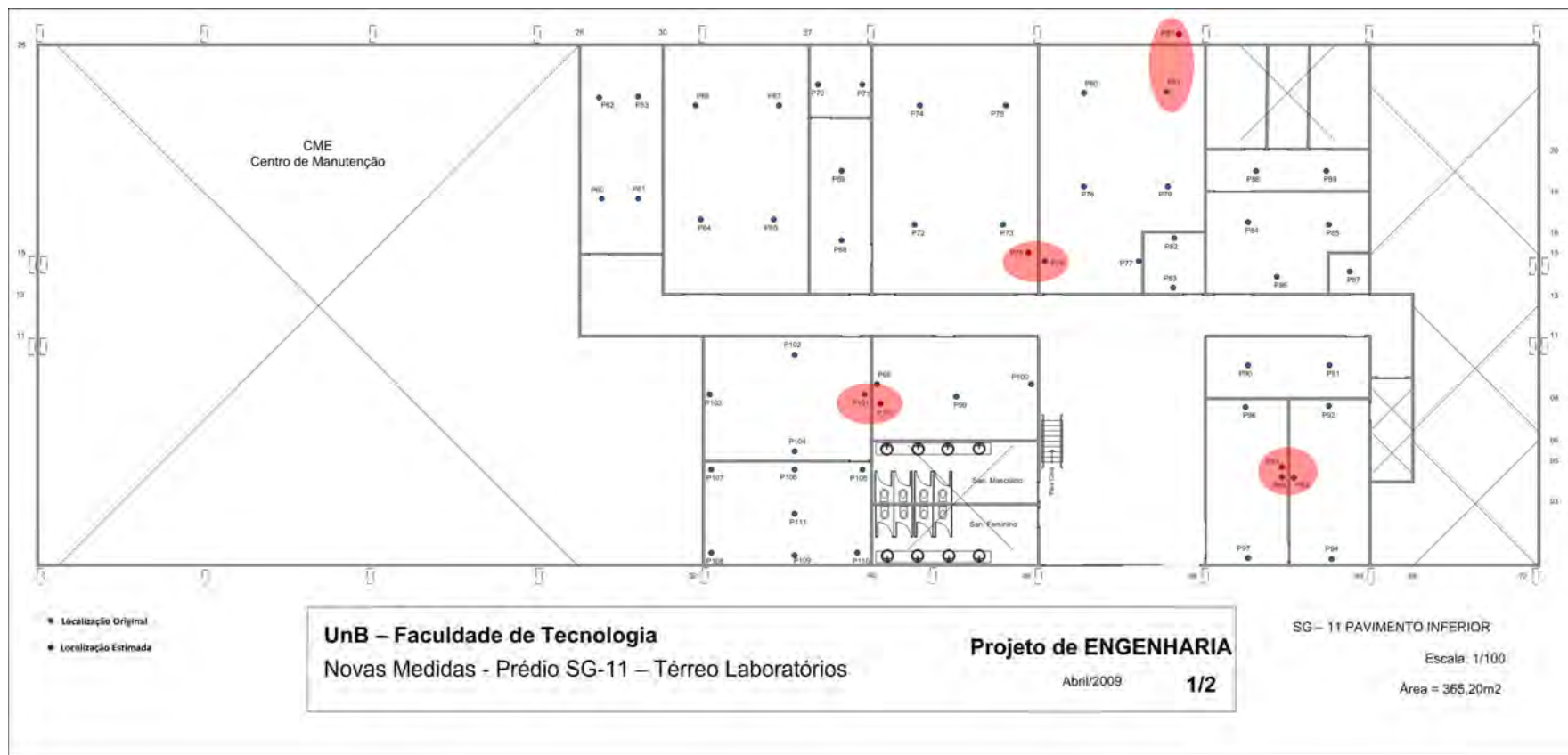


Figura 5.53 - Localização Original VS Localização Estimada pela RNA MLP – Planta Inferior – 20 Medidas – Rede MLP

### **5.3 - ALGORITMO DO VIZINHO MAIS PRÓXIMO (NEAREST NEIGHBOR - NN)**

Como etapa final dessa dissertação, foi implementado um algoritmo bastante conhecido na área de localização para que pudéssemos nos certificar das observações mais recentes em pesquisas científicas. Essas observações levam a crer que modelos neurais – principalmente aqueles baseados em redes neurais multicamadas – apresentam resultados superiores àqueles encontrados em algoritmos tradicionais. Esses resultados, geralmente são atribuídos a variáveis como tempo de simulação, capacidade superior no processamento de inúmeros pedidos de localização simultâneos e, obviamente, ganho em termos de precisão na localização.

Assim como realizado nas implementações anteriores (Rede de Kohonen e Rede MLP), iniciamos com a busca pelas variáveis de inicialização do algoritmo mais adequadas, para em seguida, simular os diversos cenários de localização e avaliar os resultados obtidos.

#### **5.3.1 - Definição das Variáveis do Algoritmo NN**

O Algoritmo NN é ligeiramente diferente das soluções anteriormente apresentadas. Trata-se de um algoritmo de fácil implementação e que necessita de poucos parâmetros para inicialização. Além disso, a fase de treinamento ocorre no mesmo instante em que se busca por um dispositivo móvel, isto é, na fase de simulação (Aprendizado Supervisionado do tipo *Lazy*). Com isso, as discussões a respeito dos “Resultados dos Treinamentos”, presentes nas redes anteriores não estarão presentes neste algoritmo.

Adotamos a medida da distância Euclidiana, e utilizamos como critério a posição de um único vizinho mais próximo (algoritmo NN simples).

### 5.3.2 - Simulações

A fase de simulação adotada para o Algoritmo NN conta com as mesmas características empregadas nos estudos anteriores. Isto é, simulação que considera a rede neural criada com todos os dados (Planta Completa) e com a simulação dividida em três etapas: Simulação com 75% dos pontos utilizados no treinamento, Simulação com pontos em regiões de fronteira e Simulação contendo novos pontos. A etapa de simulação em regiões de fronteira considerou os pontos utilizados na rede MLP pelo mesmo motivo que foi incluído na rede multicamadas (as regiões são basicamente as divisões entre os ambientes e não mais as divisões entre as classes de neurônios vencedores do mapa auto-organizável).

A saída do algoritmo NN é semelhante a da Rede Kohonen. O algoritmo rotula a amostra apresentada na simulação com o nome do ambiente associado ao arquivo de treinamento. Com isso, essa simulação avaliou erros de localização de ambientes – foco desse trabalho – e não mais a precisão da rede em metros, como ocorreu com a rede MLP.

#### 5.3.2.1 - Simulação contendo pontos utilizados no treinamento e pontos localizados em regiões de fronteira

O “treinamento” de um algoritmo NN é o armazenamento dos dados. Com isso, para as situações de “Pontos utilizados no treinamento” e “Pontos localizados em regiões de fronteira”, explorados nas redes anteriores, não se obterão erros, porque um ponto é necessariamente o seu próprio “Vizinho mais Próximo”. Com isso, os estudos com o algoritmo NN só serão explorados na situação de novos pontos (Seção 5.3.2.2).

#### 5.3.2.2 - Simulação contendo novos pontos adquiridos em novas medidas de campo aleatórias

Nessa simulação, novos pontos não presentes no arquivo de treinamento foram apresentados ao algoritmo NN e seus resultados analisados. O desempenho real do

algoritmo foi analisado por meio dos 555 pedidos de localização apresentados a ele. Os resultados encontram-se nas figuras 5.54-5.56:

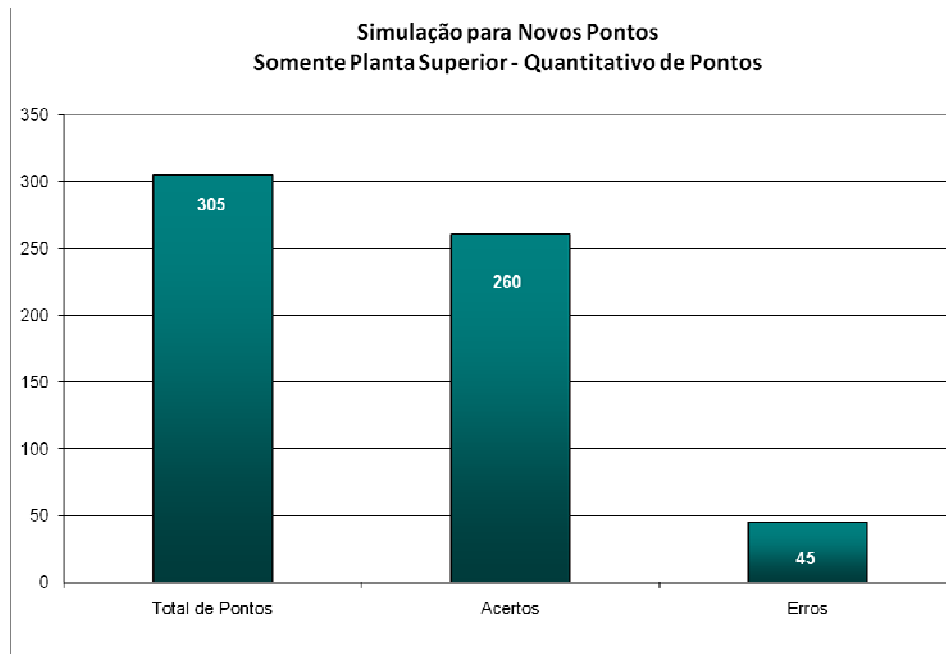


Figura 5.54 - Resultados alcançados para novos pontos - Planta Superior – Quantitativo de Pontos – NN – 85,25% de Acerto e 14,75 % de Erro

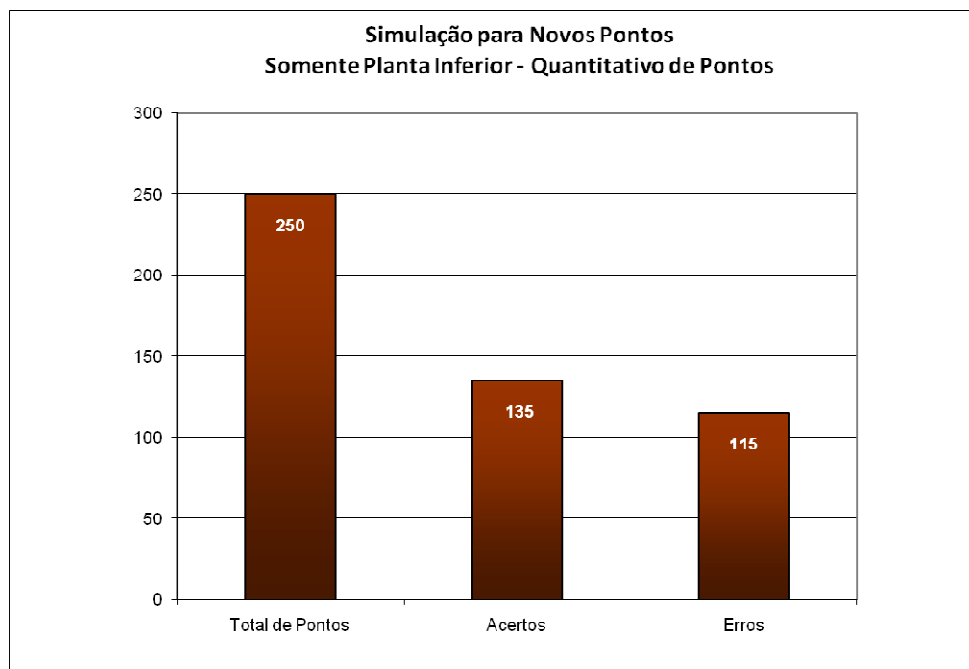


Figura 5.55 - Resultados alcançados para novos pontos - Planta Inferior – Quantitativo de Pontos – NN – 54% de Acerto e 46% de Erro

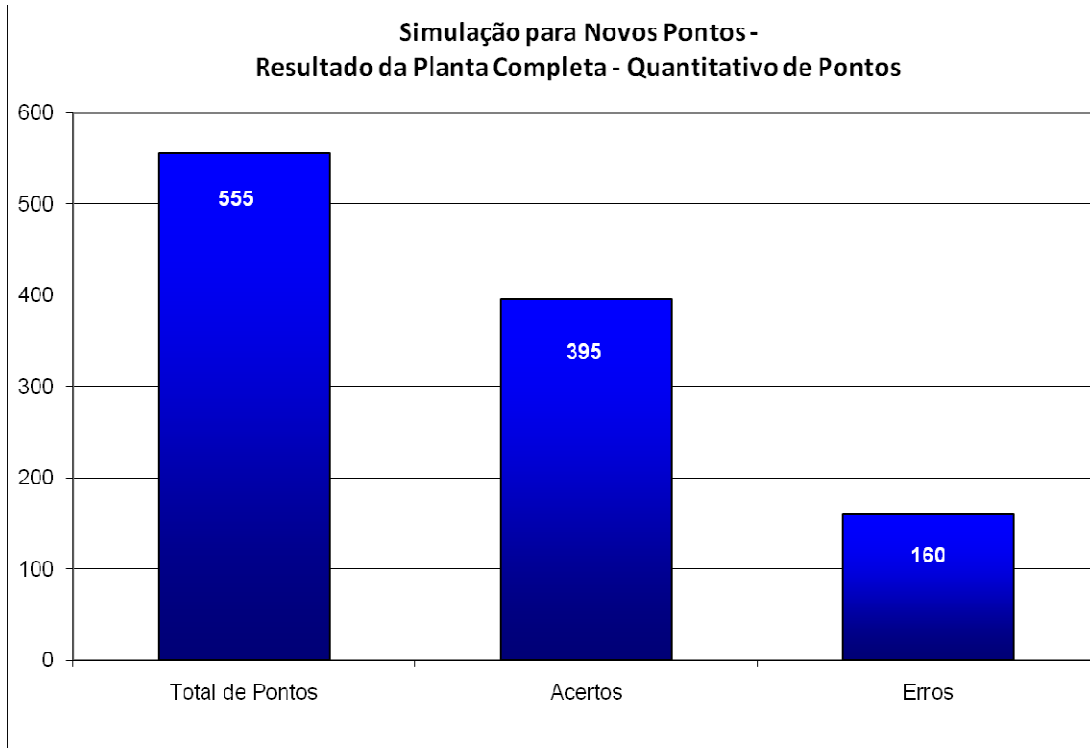


Figura 5.56 - Resultados alcançados para novos pontos - Planta Completa –  
Quantitativo de Pontos – NN – 71,17% de Acerto e 28,83% de Erro

Com esta simulação, fica mais do que clara a superioridade dos algoritmos de localização baseados em redes neurais. Para pontos não presentes no arquivo de treino, que é a situação enfrentada no dia-a-dia da tarefa de localização, o algoritmo NN não apresentou resultados satisfatórios. Pelo contrário, ao final dos 555 pedidos de localizações, a média de erro do algoritmo encontrou-se em 40%. Esses erros consistiam tanto em regiões de fronteira quanto erros dentro dos ambientes. Em muitos casos, o algoritmo informava uma posição muito diferente da posição real do móvel. Por exemplo, ocorreu situação em que o móvel encontrava-se na parte superior direita da planta e o algoritmo informou que ele encontrava na parte superior, porém, no centro da planta. Um erro muito grande para um algoritmo que é utilizado até o momento em aplicações comerciais (inclusive a aplicação de localização da Microsoft, RADAR).

Apesar da grande taxa de erro, não observamos casos de erros de localização entre andares. Esse fato não é privilegio do algoritmo, uma vez que todas as outras técnicas também não apresentaram erros. O que ocorre é que o arquivo de entrada possui



características bem distintas entre ambientes – por conta da localização dos Access points e dos valores de *RSSI* distintos mesmo em ambientes na mesma posição em andares diferentes – o que auxilia na prevenção aos erros desse tipo.

A seção seguinte compara os três algoritmos em termos de precisão, tempo de processamento e de simulação, total de iterações, entre outros.

#### **5.4 - ANÁLISES COMPARATIVAS: REDES NEURAIIS DE KOHONEN, PERCEPTRON MULTICAMADAS (MLP) E ALGORITMO DO VIZINHO MAIS PRÓXIMO (NEAREST NEIGHBOR - NN)**

Em uma análise preliminar de todos os resultados passados, podemos concluir que os três métodos alcançaram resultados interessantes e que, certamente, a técnica de inteligência artificial aplicada a esse tipo de localização é confiável (em detrimento ao algoritmo NN). Esse fato é reforçado pelos diversos estudos, trabalhos de mestrado, doutorado e publicações em revistas científicas citados na revisão bibliográfica desse trabalho.

Com isso, essa seção fará comparações entre todos os métodos apresentados anteriormente com base nos seguintes aspectos:

1. Análise de erros de localização nas medidas para a simulação contendo 75% dos pontos originais utilizados no treinamento das redes;
2. Análise de erros de localização nas regiões de fronteiras de ambientes/classes;
3. Análise de erros de localização para novas medidas adquiridas experimentalmente e não presentes no treinamento das redes;
4. Total de Iterações na fase de Treinamento;
5. Tempo de Execução dos Algoritmos de Treinamento;
6. Tempo de Execução dos Algoritmos baseado na quantidade de pedidos de localização.

### 5.4.1 - Simulação contendo 75% dos pontos originais

As duas redes neurais forneceram resultados bastante confiáveis, onde o maior percentual de erro identificado ficou na faixa de 3%. Foram utilizadas 690 medidas do arquivo original de treinamento, para assim, termos 75% dos pontos de cada ambiente.

A seguir, temos a comparação gráfica entre as Redes de Kohonen, MLP e o Algoritmo NN.

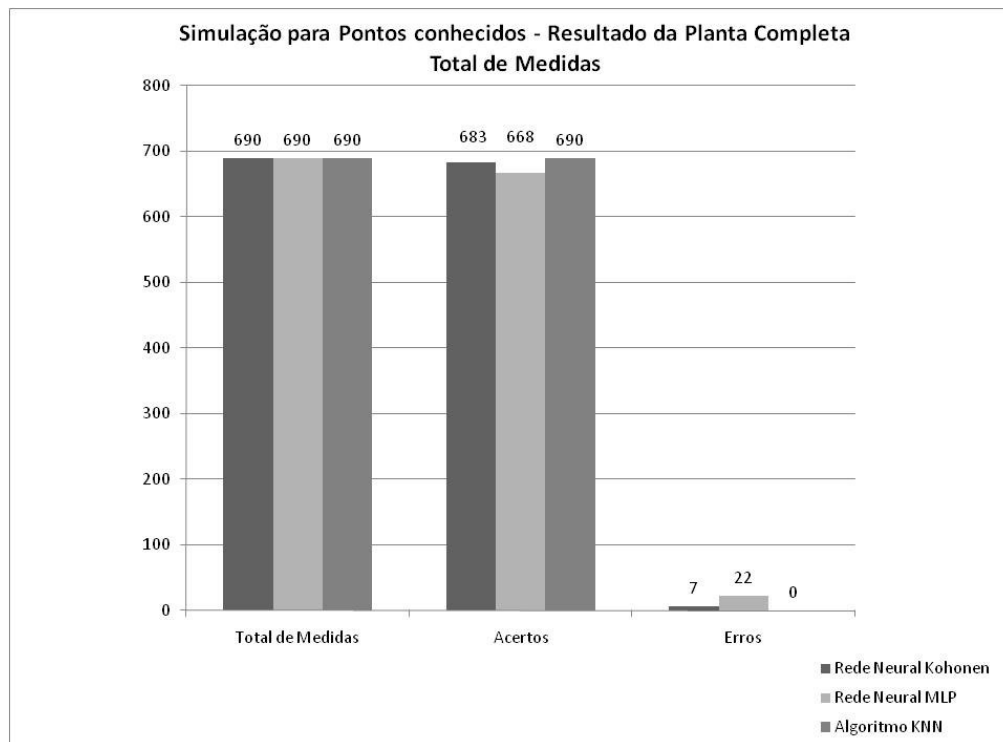


Figura 5.57 - Comparação do resultado da Simulação para Pontos conhecidos – Redes Kohonen, MLP e Algoritmo NN

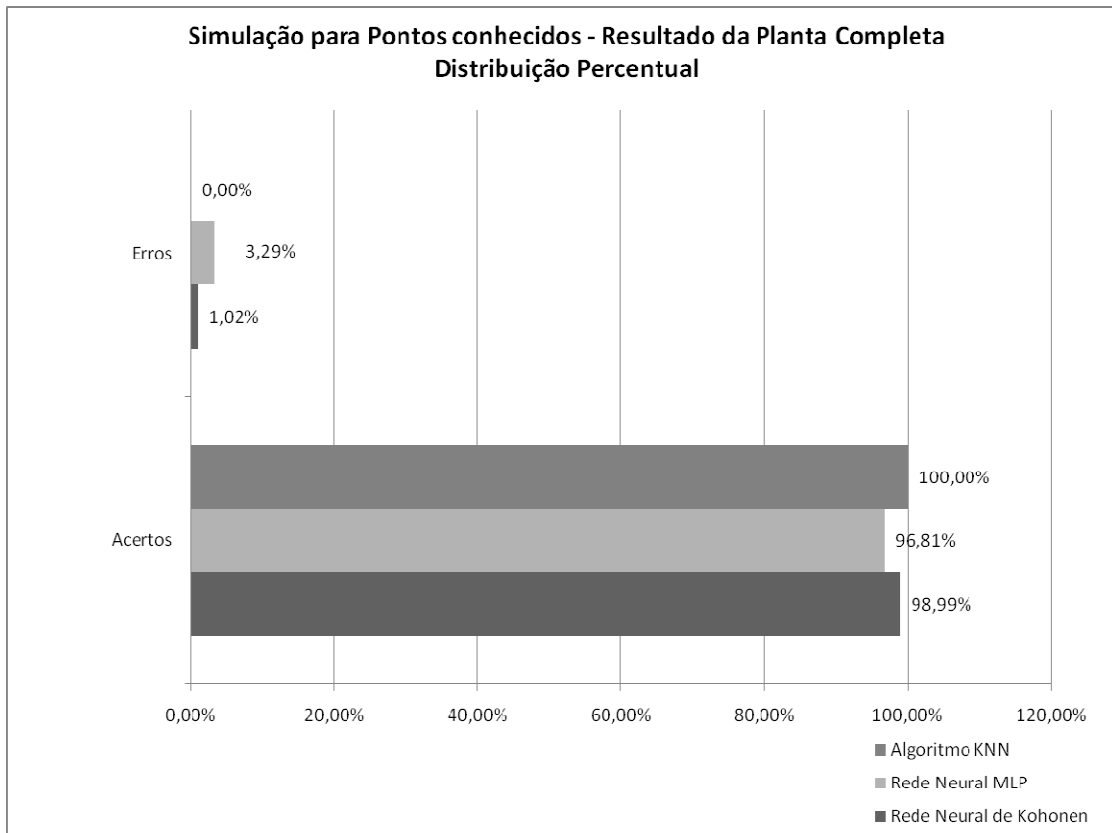


Figura 5.58 - Comparação do resultado percentual da Simulação para Pontos Conhecidos – Redes Kohonen, MLP e Algoritmo NN

Como não há a possibilidade real de comparação com o Algoritmo NN, nesse instante teremos somente considerações a respeito dos algoritmos neurais.

Apesar das duas redes terem resultados idênticos quanto aos erros de localização entre andares, a situação é diferenciada para os outros resultados. Nesse caso, é nítida a diferença de resultados entre a Rede Kohonen e MLP. A primeira delas apresentou resultados muito próximos a 1% de erro (somente 7 erros de localização) enquanto a rede MLP errou 22 vezes, gerando um erro total de 3,29%.

A Rede Kohonen apresentou resultados 3 vezes mais acurados que a Rede MLP. A Rede MLP ainda possui a desvantagem de apresentar grandes erros dentro de um mesmo ambiente. Existiram situações em que o desvio da localização chegou a 4 metros. Apesar dessa informação de precisão em metros possuir fins informativos, uma vez que o objetivo do trabalho é localizar a região onde se localiza o dispositivo móvel, é

interessante ressaltar essa característica como forma de explicar a forma com que a rede MLP respondeu as simulações em regiões de fronteira.

Mesmo a Rede de Kohonen não possuindo como saída uma coordenada geográfica, o que possibilitaria o confronto da precisão das redes em metros, é suficiente o comportamento dessa rede em regiões que dividem ambientes. Em caso de erros grandes nessas regiões, pode-se inferir que a rede apresenta a mesma deficiência da rede MLP quanto aos deslocamentos de pontos em um ambiente.

Os estudos comparativos a seguir ilustram essa situação.

#### **5.4.2 - Simulação para as Regiões de Fronteira**

Há de se ter cuidado com as comparações para este tipo de simulação. Como sabemos, a Rede de Kohonen trabalhou com menos amostras que a Rede MLP. Os gráficos individuais que ilustram o resultado de cada rede indicam grande vantagem para a Rede de Kohonen. A rede de Kohonen trabalhou com 270 medidas e não apresentou erros (100% de acurácia), enquanto que a rede MLP trabalhou com 440 medidas e errou em 50 delas (88,64% de acurácia). O Algoritmo NN utilizou o mesmo número de amostras da rede MLP.

Os gráficos 5.59 e 5.60 consolidam estes resultados:

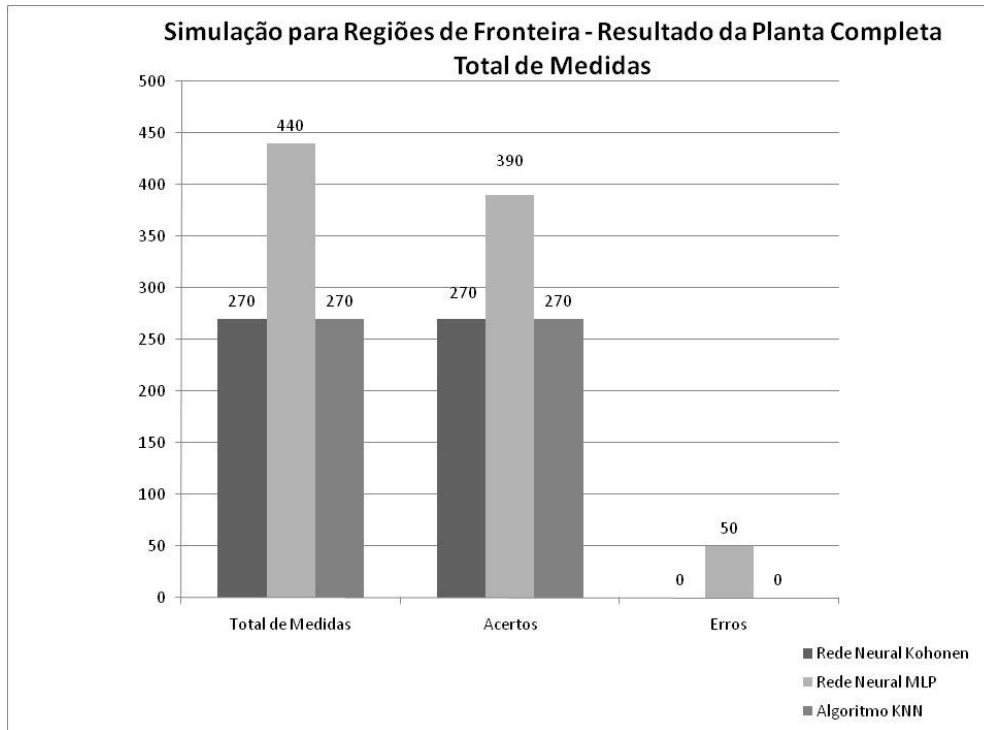


Figura 5.59 - Comparação do resultado da Simulação em Regiões de Fronteira – Redes Kohonen, MLP e Algoritmo NN

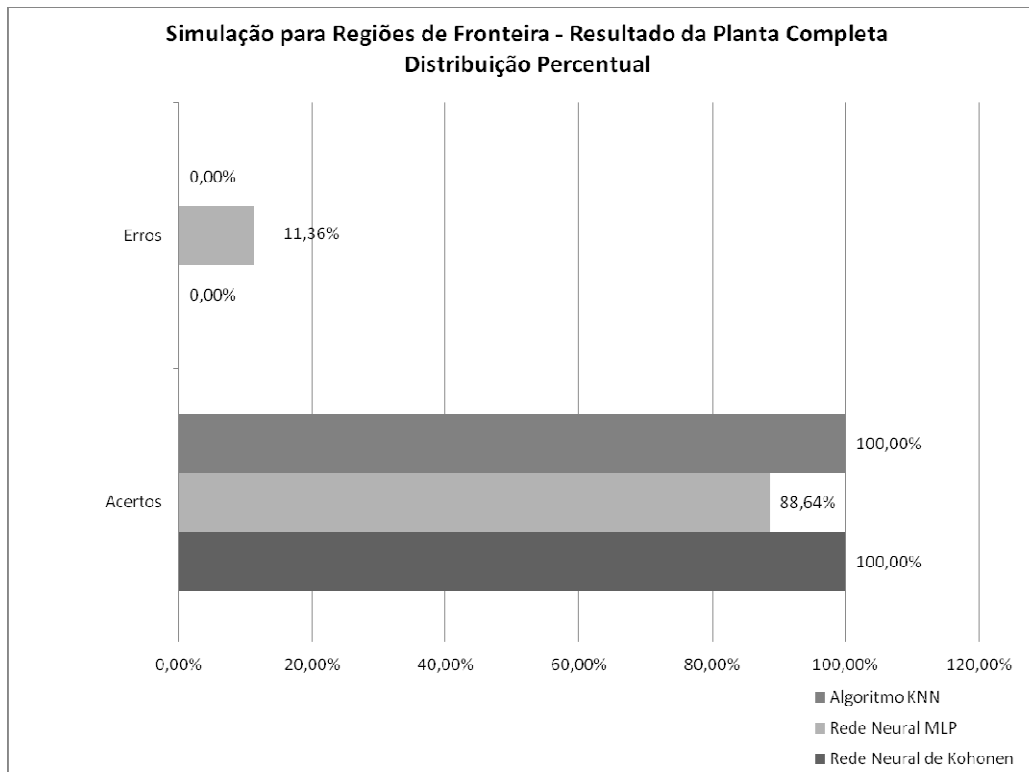


Figura 5.60 - Comparação do resultado percentual da Simulação em Regiões de Fronteira – Redes Kohonen, MLP e NN

Assim como na seção anterior, os comentários se resumirão as redes neurais.

Mais uma vez, a Rede de Kohonen obteve o melhor resultado, mostrando também que, atua de maneira bastante satisfatória em regiões que separam classes/ambientes. Como a Rede MLP apresenta variações de precisão geográfica dentro dos ambientes, partes desses erros foram classificados por poucos metros ou até centímetros fora do local correto.

No que tange a erros de localização entre andares, a situação permanece. As duas redes apresentaram um aproveitamento de 100%.

### 5.4.3 - Simulação para novas medidas experimentais

Essa simulação teve como objetivo avaliar a capacidade das redes em generalizar, isto é, partindo de dados de entrada nunca vistos nas fases de treinamento encontrar uma classe (rede Kohonen) com características semelhantes ou a melhor região que descreve a localização do móvel (rede MLP) para novas medidas experimentais. Além disso, é neste tipo de simulação que o algoritmo NN pode ser efetivamente comparado com as demais técnicas de localização. Essas medidas representam 50% das originais de cada ambiente, totalizando, 555 medidas.

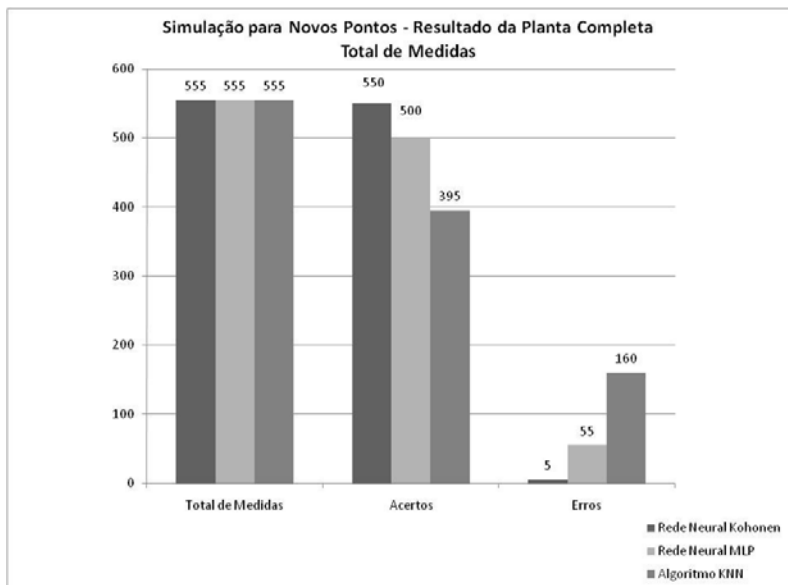


Figura 5.61 - Comparação do resultado da Simulação para novas medidas – Redes Kohonen, MLP e Algoritmo NN

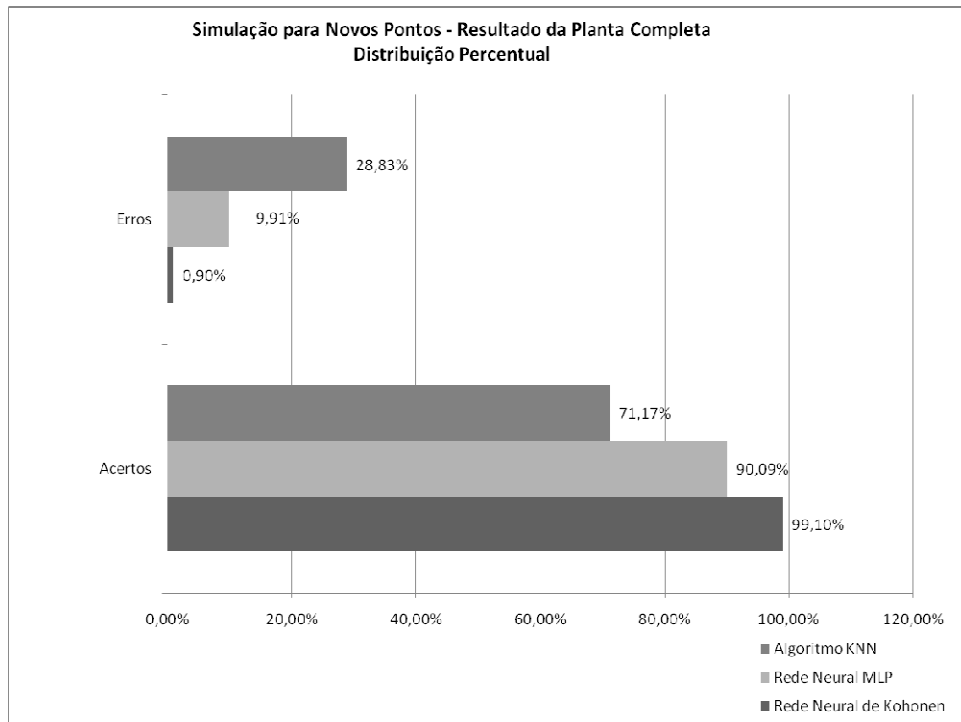


Figura 5.62 - Comparação do percentual do resultado da Simulação para Novas Medidas – Redes Kohonen, MLP e Algoritmo NN

A Rede de Kohonen, não muito diferente do que ocorria nas outras simulações analisadas, apresentou resultados bastante superiores a rede MLP e nesse último caso, superior ao algoritmo NN. Para esse caso, a aplicação dessa técnica resultou num aumento de acurácia de 11x para a rede MLP e 40x para o algoritmo NN. Enquanto o algoritmo NN errou 160 vezes, a rede MLP 55 das vezes, Kohonen errou em 5 delas.

Para este experimento, os resultados ficaram próximos de 0% de erro, 11% de erro e 40% para as redes Kohonen, MLP e algoritmo NN, respectivamente. Com isso, fica clara a viabilidade do uso da rede Kohonen no trabalho de localização *indoor*.

Os próximos aspectos a serem analisados serão: Total de Iterações no processo Treinamento das Redes, Tempo de Execução e Tempo de Execução dos Algoritmos baseado na quantidade de pedidos de localização

#### 5.4.4 - Total de Iterações na fase de Treinamento

Iteração em um algoritmo de treinamento é o momento em que todos os dados de entrada são apresentados para uma rede neural. Quanto maior a quantidade de iterações, maior será o número de vezes que a rede irá analisar e “aprender” com esses mesmos dados. Dependendo da complexidade do problema, um conjunto de iterações pode levar minutos ou até mesmo horas de execução.

Os gráficos apresentados a seguir ilustram a quantidade de iterações necessárias até que cada rede neural apresentasse o melhor resultado e assim, definidos os parâmetros adequados para o treinamento.

Como o algoritmo NN não possui um “treinamento” propriamente dito, portanto, sendo diferente das redes MLP e Kohonen, ele constará nos gráficos com o valor zero.

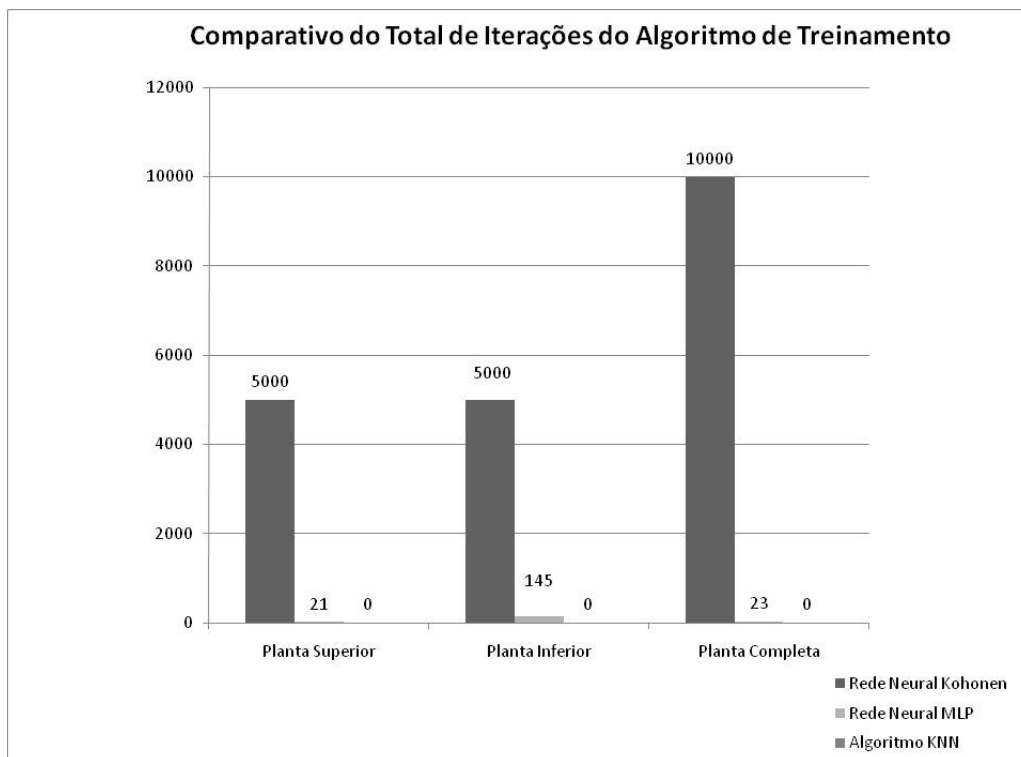


Figura 5.63 - Comparativo do Total de Iterações do Algoritmo de Treinamento – Redes Kohonen, MLP e Algoritmo NN

Podemos observar que nem sempre um maior número de iterações é garantia de que a rede neural trará resultados com maior precisão. A Rede MLP convergiu em menos iterações quando comparada com a Rede Kohonen. Entretanto, essa mesma rede (MLP)



foi aquela que apresentou os piores resultados nas comparações. Isto é, apesar da rede MLP ter apresentado resultados com acurácia superior a 85% com menos iterações, a rede Kohonen, que necessitou de uma longa quantidade de iterações para esta fase, errou menos e demonstrou ser mais precisa (média de 95% de eficiência). Em alguns casos, a Rede de Kohonen reduziu o erro em 11x, comparada com a Rede MLP.

Como o procedimento de treino é realizado uma única vez, é de se considerar que, mesmo com a quantidade de iterações elevadas para a Rede Kohonen, o aumento significativo da precisão advinha nesse processo compensa a diferença de iterações no treino entre um tipo de rede e outro.

#### **5.4.5 - Tempo de Execução dos Algoritmos de Treinamento**

Em relação a medidas de tempo, a próxima comparação é relativa ao tempo de execução dos algoritmos, medida intimamente relacionada com a quantidade de iterações necessárias para que a rede neural traga resultados confiáveis. Cabe ressaltar de antemão que existe uma diferença entre avaliar tempo de execução de um algoritmo e tempo de processamento.

De maneira geral, o tempo de processamento é uma medida mais complexa e de bastante precisão, normalmente associada à quantidade de operações de ponto flutuante por segundo (FLOPS).

Essa medida é utilizada para determinar o desempenho de um computador, especificamente no campo de cálculos científicos, que fazem grande uso de cálculos com ponto flutuante, similar as instruções por segundo [76]. Para que os FLOPS sejam úteis como unidade de medida de ponto-flutuante, um *benchmark* deve estar disponível em todos os computadores de interesse. Existem diversos fatores no desempenho do computador para medir a velocidade do cálculo de números pontos-flutuantes, como a desempenho de Entrada/Saída, comunicação do interprocessador, coerência de cache, entre outros. Todos esses fatores são relacionados com a medida de processamento.

No estudo em questão, a análise de tempo se resumiu a medir o horário de início e término da execução do algoritmo de treinamento. Lembramos que a medida de tempo para o algoritmo NN será avaliada na seção a seguir (Tempo de Execução dos Algoritmos baseado na quantidade de pedidos de localização). O gráfico comparativo está na figura 5.64.

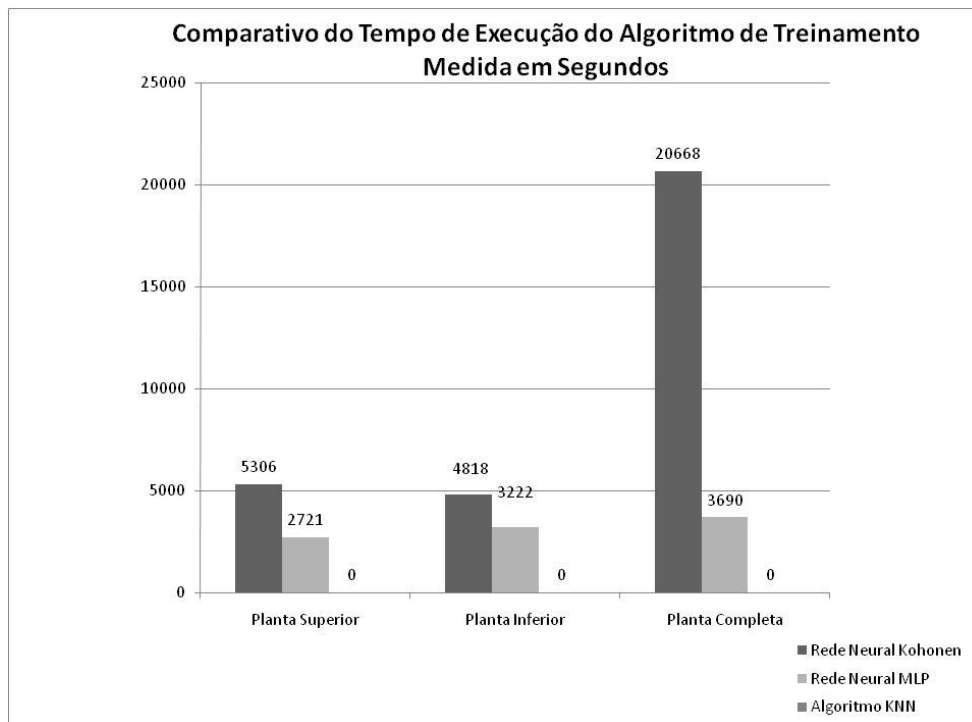


Figura 5.64 - Comparativo do Tempo de Execução do Algoritmo de Treinamento das Redes Neurais Kohonen, MLP e Algoritmo NN

Conforme falamos na seção anterior, apesar da Rede MLP ter exigido um número menor de iterações e ter convergido em menos tempo de execução, a Rede de Kohonen apresentou uma precisão maior para todas as simulações.

Com isso, e seguindo a mesma linha de raciocínio já delineada anteriormente, a Rede Kohonen ainda apresenta vantagem no quesito tempo de execução de treinamento quando analisado sob o prisma da precisão alcançada.

#### **5.4.6 - Tempo de Execução dos Algoritmos baseado na quantidade de pedidos de localização**

Por fim, a última medida de comparação entre algoritmos é o tempo de execução baseada na quantidade de pedidos de localização. Um algoritmo de localização, além de apresentar a precisão desejada, deve trabalhar satisfatoriamente com pedidos de localização simultâneos ou não, e em grande escala. Essa comparação possibilitará a avaliação do algoritmo NN em mais detalhes.

Os gráficos apresentam valores partindo de 10 a 650 localizações, que é basicamente o total de pontos utilizados em cada etapa das simulações executadas. Como os resultados do algoritmo NN destoaram em muito dos modelos neurais, criamos 2 gráficos e uma tabela para facilitar o entendimento.

O primeiro gráfico (figura 5.65) exhibe a comparação com os três métodos desenvolvidos nesse trabalho. O segundo gráfico (figura 5.66), detalha o tempo de execução dos modelos neurais (Kohonen e MLP).

A tabela 5.16 consolida as informações dos dois gráficos.

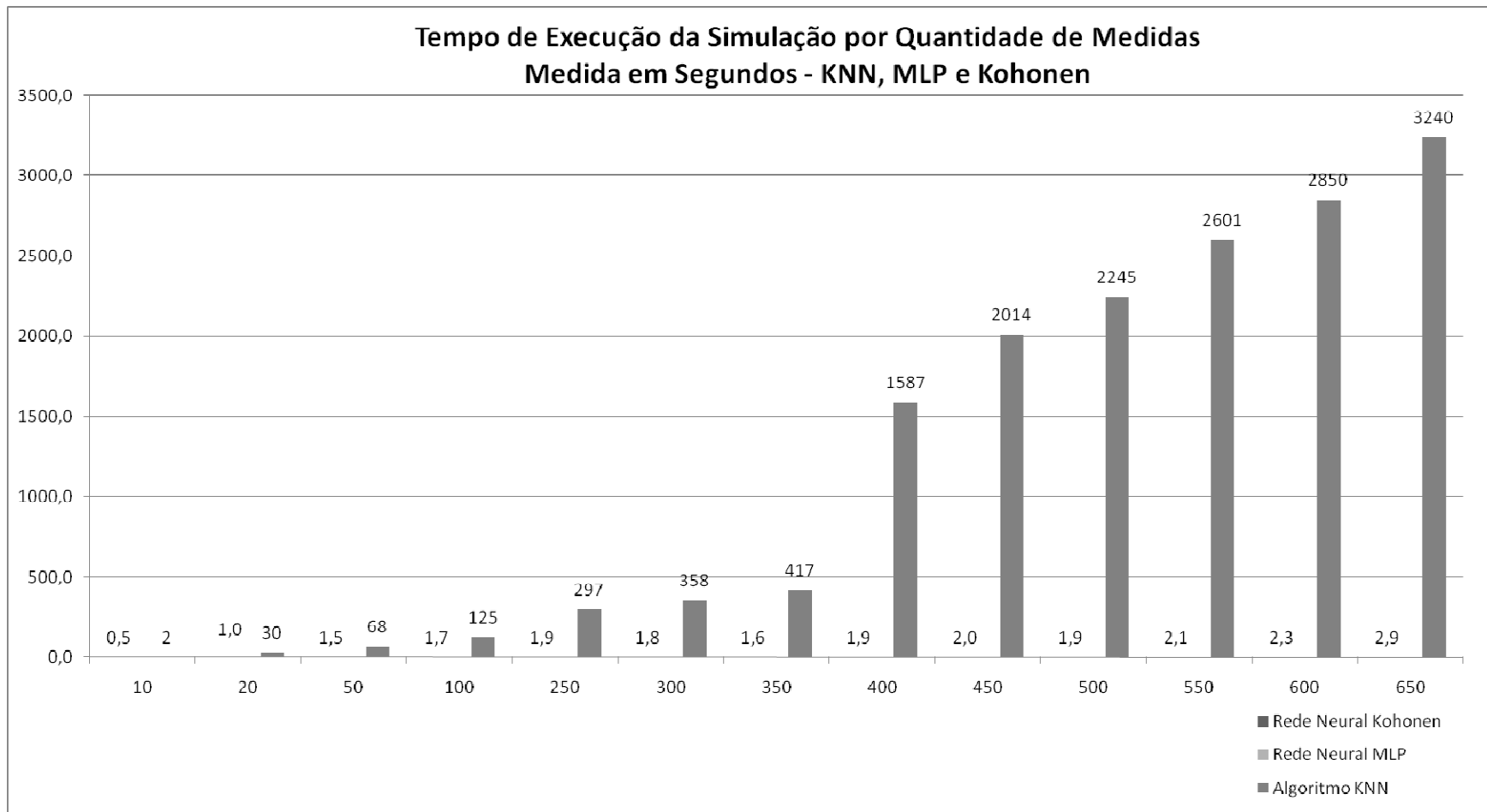


Figura 5.65 - Tempo de Execução da Simulação por Quantidade de Medidas – Algoritmo NN, Redes Neurais MLP e Kohonen

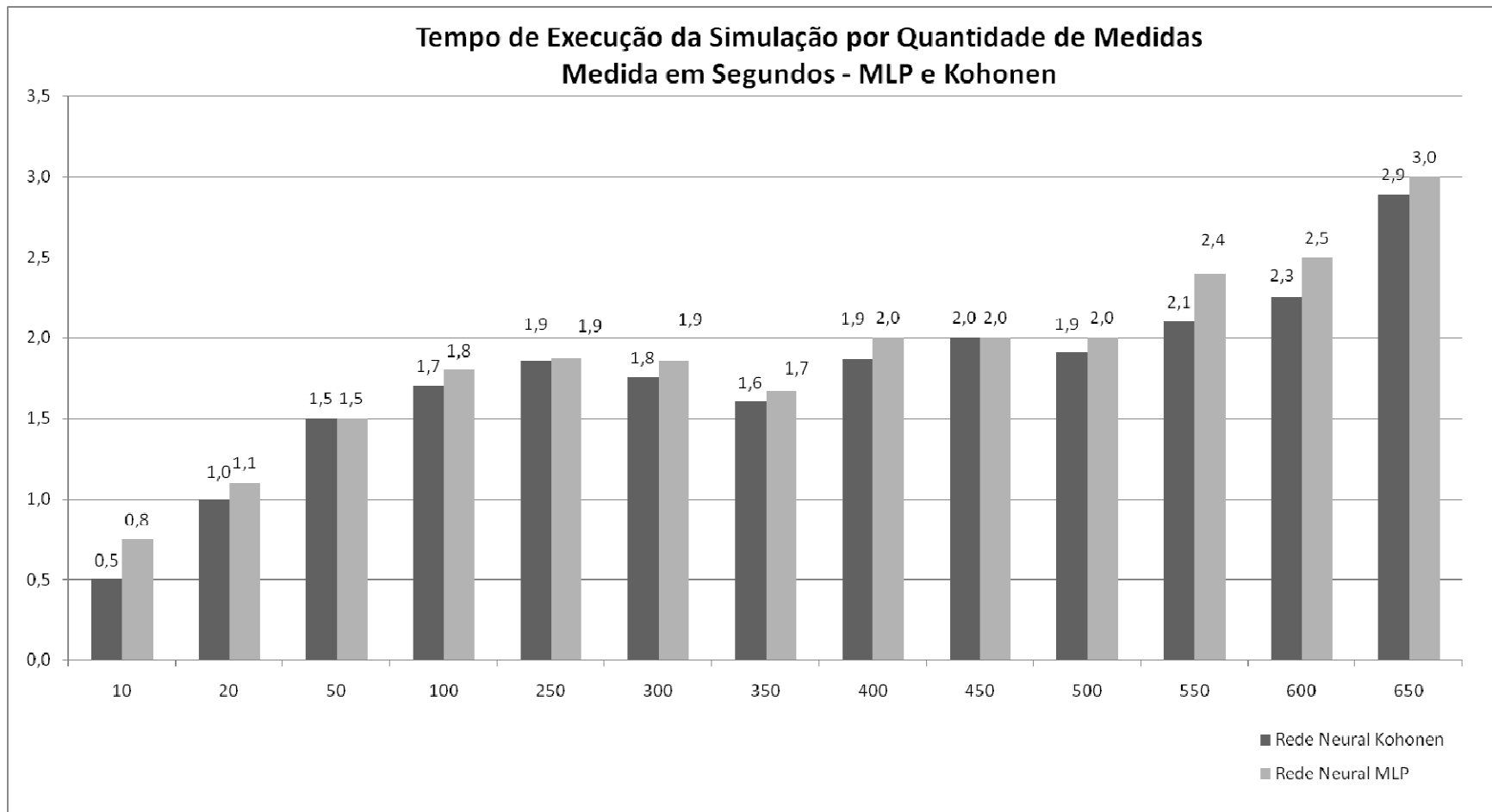


Figura 5.66 - Tempo de Execução da Simulação por Quantidade de Medidas –Redes Neurais MLP e Kohonen

Tabela 5.16 - Tempo de Execução das Simulações – Algoritmo NN, Redes Neurais MLP e Kohonen

Tempo de Execução da Simulação (Em Segundos)			
Pedidos	Kohonen	MLP	NN
10	0,5	0,8	2
20	1,0	1,1	30
50	1,5	1,5	68
100	1,7	1,8	125
250	1,9	1,9	297
300	1,8	1,9	358
350	1,6	1,7	417
400	1,9	2,0	1587
450	2,0	2,0	2014
500	1,9	2,0	2245
550	2,1	2,4	2601
600	2,3	2,5	2850
650	2,9	3,0	3240

É nítida a diferença entre as redes neurais e o algoritmo NN no tempo de execução das simulações. Enquanto o maior tempo encontrado para modelos neurais era de 3 segundos (Rede MLP), para a mesma situação (650 localizações), o algoritmo NN levou 3240 segundos para informar a localização desses móveis. Entretanto, para localizações com poucos pontos (10 móveis), a rede NN demorou 2 segundos para fornecer o dado. Apesar de 2 segundos ser 4 vezes o tempo necessário para a RNA Kohonen ainda é um tempo aceitável de resposta.

Contudo, os resultados do algoritmo NN são inaceitáveis para esse tipo de experimento, uma vez que a aplicação necessita de respostas rápidas sem que a capacidade computacional seja comprometida. O grande problema do algoritmo NN é que apesar de requerer pouco esforço durante a etapa “de treinamento”, necessita de um alto custo computacional para rotular um novo exemplo, pois, no pior dos casos, esse exemplo deverá ser comparado com todos os exemplos contidos no conjunto de exemplos de treinamento. Certamente é o que ocorreu com o aumento das amostras.

É conhecido que existem grandes possibilidades computacionais na atualidade para amenizar o problema, como o processamento paralelo ou a computação em grade. Mas não há muito sentido paralelizar uma aplicação que naturalmente sofre de deficiências com grandes amostras e no caso de pontos desconhecidos no arquivo de treino – que é a

principal aplicação real do algoritmo – não obtém resultados próximos de 1% de erro (como os modelos neurais). Com isso, para que este algoritmo apresente resultados melhores, deve possuir um arquivo de treino muito extenso, com muitas medidas próximas umas das outras, de modo que se aumente a possibilidade de uma simulação encontrar um vizinho no arquivo de treinamento com características muito semelhantes. Isso, além de ser desnecessário, sabemos que é inviável.

Em se tratando das Redes MLP e Kohonen, observamos resultados semelhantes, porém ligeiramente favoráveis a rede Kohonen por frações de segundos. Esses resultados só reforçam a tese de que modelos baseados em Redes Neurais são mais adequados a localização em detrimento dos métodos tradicionais atualmente empregados nesse problema. De maneira especial, mostramos que a Rede Neural de Kohonen é capaz de apresentar resultados mais sólidos e confiáveis, devendo ser considerado como uma nova solução para o difícil problema da localização *indoor*.

## 6 - CONCLUSÕES

Neste trabalho, verificou-se que o método das Redes Neurais de Kohonen pode ser utilizado para a localização de dispositivos móveis em um ambiente fechado, e para trazer uma nova maneira de se analisar o comportamento de algoritmos de localização em termos de precisão, velocidade de convergência e flexibilidade. Neste trabalho, a informação da intensidade do sinal recebido (*RSSI*) de adaptadores sem fio foi utilizada como base para a localização de dispositivos móveis por meio de 3 técnicas distintas: Algoritmo do Vizinho Mais Próximo (NN), Redes Neurais Artificiais Percetron Multicamadas e Redes Neurais de Kohonen. Foram realizadas uma série de simulações com esses algoritmos tomando como base três cenários distintos: Pontos Conhecidos, Pontos localizados em Regiões de Fronteira e Novos Pontos. Com isso, pode-se então chegar a algumas conclusões:

- Com a Rede Neural de Kohonen é possível simular a localização de um dispositivo em ambiente fechado desde que os *APs* escolhidos para compor o mapa de propagação utilizado no treinamento tenham a capacidade de cobertura em toda a planta do ambiente. Isso evitará inconsistências na rede resultantes do uso de *APs* que cumpram parcialmente esse quesito.
- O processo de medidas experimentais para compor o mapa de propagação é importante para a determinação da convergência e precisão dos algoritmos de localização. Quando as medidas são inseridas com ruído ou erro de mensuração, é esperado prejuízo no desempenho do treinamento e aumento no número de erros nos pedidos de localização.
- Ainda em relação ao processo de medidas, é interessante que se tenha uma distribuição de pontos por ambiente favorável e que a mensuração do *RSSI* em cada um deles seja realizada mais de uma vez e em intervalos constantes de tempo.



- O uso de medidas de emissores em andares diferentes foi importante na formação do Mapa de Kohonen com maior acurácia e em menor tempo. Fato este baseado em testes.
- O Algoritmo NN só apresenta resultados satisfatórios quando o móvel encontra-se num ponto conhecido no seu arquivo de referência. Isso é uma grande desvantagem quando comparada a técnica de redes neurais que é capaz prover a localização de um móvel mesmo que seja em um ponto não conhecido do mapa de treinamento (generalização).
- O método de Redes Neurais mostrou-se mais eficiente em todos os cenários apresentados. Quando comparados RNA Perceptron e RNA Kohonen, este último obteve ganhos significativos nos casos de Simulação para Pontos Conhecidos (aumento de precisão de 3x) e Regiões de Fronteira (100% de acerto contra 88,64% da RNA MLP).
- Do ponto de vista do tempo de execução de treinamento dos algoritmos, Kohonen foi aquele que apresentou os resultados menos favoráveis. Porém, o fato é aceitável uma vez que a referida fase é realizada uma única vez.
- A capacidade e tempo de resposta a pedidos de localização são de extrema importância para o algoritmo. Nesse quesito, a RNA Kohonen não só provou que é o mais preciso, como aquele que apresentou os menores tempos de resposta mesmo para grandes quantidades de pedidos simultâneos (650 solicitações).
- No cenário de novos pontos, a RNA Kohonen apresentou aumento de acurácia de 40x em relação ao Algoritmo KNN e 11x em relação a RNA MLP.
- Apesar das diferenças de desempenho entre os algoritmos, um ponto comum entre eles é que nenhum deles, em nenhum cenário, apresentou erros de localização entre andares.

- O presente trabalho foi publicado na forma de artigo em congressos nacionais referenciados em [47,48].
- Por fim, percebeu-se que, a aplicação da RNA de Kohonen ao problema nos trouxe resultados mais precisos, em menor tempo e com uma maior capacidade de processamento de solicitações simultâneas.

## 6.1 SUGESTÕES PARA PESQUISAS FUTURAS

As pesquisas que utilizam redes neurais artificiais para localização *indoor* estão em pleno desenvolvimento. Isto pode ser comprovado pela quantidade significativa de artigos recentes que utilizam o método. Esta dissertação representa o primeiro passo em direção à aplicação de RNAs do tipo Kohonen como forma de solução ao problema e, ao mesmo tempo, um novo tema de pesquisa dentro do Programa de Pós-Graduação em Engenharia Elétrica da Universidade de Brasília, que no futuro poderá abordar temas como:

- Estudar a influência do aumento da quantidade de *APs* para alimentar o mapa de Kohonen e os possíveis ganhos em termos de precisão.
- Definição de método que identifique o número ideal de *APs*, evitando o processamento de dados desnecessários na fase de treinamento. Com isso, é possível que se obtenha ganhos na velocidade do algoritmo na fase de treinamento.
- Utilização do modelo proposto como base para aplicações de computação ubíqua, tais como a localização de possíveis clientes dentro de um *shopping center*, envio de ofertas baseadas na proximidade de loja, monitoramento de cargas/bens dentro de ambientes, entre outros.
- Extensão desse trabalho no sentido de comparação com outros métodos de localização emergentes.

- Avaliação do uso de medidas ruidosas nos processos de treinamento e simulação da Rede Neural de Kohonen.
- Realização de medidas de intensidade de campo com trânsito de pessoas e objetos.
- Comparação da acurácia entre o algoritmo de Kohonen com a técnica de *C-Means Clustering*.
- Avaliação do desempenho do algoritmo de Kohonen na situação de adição de novos locais ao mapa criado.

## REFERÊNCIAS BIBLIOGRÁFICAS

- [1] AGGARWAL, C. C., HINNEBURG, A., e KEIM, D.A., “On the surprising behavior of distance metrics in high dimensional space”. In Lecture Notes in Computer Science, Springer, 2001.
- [2] AGRAWAL D.P., ZENG Q., "Introduction to Wireless and Mobile Systems", Thomsom Brooks/Cole, USA, 2003.
- [3] AHA, D.W., KIBLER D., ALBERT, M.K., “Instance-based learning algorithms”. Machine Learning 6, 1991.
- [4] ALPAYDIN, E., “Introduction to Machine learning”. MIT Press, Cambridge - MA, England, 2004.
- [5] AOKI H., SCHIELE B., PENTLAND A., “Realtime personal positioning system for wearable computers” In International Symposium on Wearable Computers ISWC’99, October 1999.
- [6] AZUMA R., “Tracking requirements for augmented reality” Communications of the ACM, 36(7), July 1997.
- [7] BAHL P., PADMANABHAN V. N., “RADAR: An in-building RF-based user location and tracking system” In IEEE Infocom 2000, 2000.
- [8] BAHL P., PADMANABHAN V. N., BALACHANDRAN A., “Enhancements to the RADAR user location and tracking system” Technical Report MSR-TR-00-12, Microsoft Research, February 2000.
- [9] BATTITI, R., “Location-aware computing: A Neural network model for determining location in wireless LANs”, University degli Studi di Trento, Tech. Rep. DIT-0083, Feb. 2002.

[10] BATTITI, R., VILLANI A., LE NHAT T., "Neural Network models for intelligent networks: deriving the location from signal patterns", in proceedings of AINS2002,UCLA,May 2002.

[11] BATTITI R., VILLANI A., LE NHAT T., "Neural Network models for intelligent networks: deriving the location from signal patterns", in Proceedings of AINS2002, UCLA, May 2002.

[12] BATTITI R., NHAT T. L., VILLANI A. "Location-aware computing: a neural network model for determining location in wireless LANs" Technical Report Technical Report DIT-02-0083, 2002.

[13] BERNA M., LISIEN B., SELLNER B., GORDON G., PFENNING F., THRUN S., "A learning algorithm for localizing people based on wireless signal strength that uses labeled and unlabeled data" In 8th International Joint Conference on Artificial Intelligence (IJCAI'03), Acapulco, Mexico, August 2003.

[14] CASTRO P., CHIU P., KREMENEK T., MUNTZ R., "A probabilistic room location service for wireless networked environments", In Ubicomp 2001: Ubiquitous Computing. Springer, 2001.

[15] CASTRO P., MUNTZ R., "Managing context for smart spaces" IEEE Personal Communications, October 2000.

[16] CHAPELLE, O., SCHÖLKOPF, B., e ZIEN, A., "Semi-Supervised Learning". MIT Press, Cambridge, MA, 2006.

[17] CHEN G., KOTZ D., "A survey of context-aware mobile computing research" Technical Report Dartmouth Computer Science Technical Report TR2000-381, 2000.

[18] CHRIST T., GODWIN P., "A prison guard duress alarm location system" In IEEE International Carnahan Conference on Security Technology, March 2000.

- [19] CRANE R. K., "Propagation Handbook for Wireless Communication System Design", CRC Press, New York, 2003.
- [20] CROW B. P., WAJAJA I., KIM J.G., SAKAI P.T., "IEEE 802.11 Wireless Local Area Networks", IEEE Communications Magazine, London,UK.
- [21] ELNAHRAWY E., LI X., MARTIN R. P., "The limits of Localization using signal strength: A comparative study", In Proceedings of the first IEEE international conference on Sensor and Ad Hoc Communications and Networks (SECON 2004), Santa Clara CA, Oct 2004.
- [22] ENGE P., MISRA P., "Special issue on GPS: The global positioning system" Proceedings of the IEEE, January 1999.
- [23] ETSI, "HIPERLAN Function Specification", ETSI Draft Standard July, 1995.
- [24] FAU, Faculdade de Arquitetura e Urbanismo, Universidade de Brasília, URL: <http://www.unb.br/fau/guia/engenhariaeletrica.htm>, Acessado em Julho de 2009.
- [25] FCC, Federal Communications Commission Report and Order 96-264. "Revision of the commission's rules to ensure compatibility with enhanced 911 emergency call systems". Technical Report Docket No. 94-102, FCC, July 1996.
- [26] FERRERO, C.A., *Algoritmo kNN para previsão de dados temporais: funções de previsão e critérios de seleção de vizinhos próximos aplicados a variáveis ambientais em limnologia*. Instituto de Ciências Matemáticas e de Computação – ICMC-USP, Dissertação de Mestrado, Universidade de São Paulo – (USP - São Carlos), 2009.
- [27] GELLERSEN H.W., SCHMIDT A., BEIGL M., "Adding some smartness to devices and everyday things" In IEEE Workshop on Mobile Computing Systems and Applications, December 2000.
- [28] GRUTESER M., GRUNWALD D., "Enhancing location privacy in wireless lan through disposable interface identifiers: A quantitative analysis" In First ACM

International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots, September 2003.

[29] HARTER A., HOPPER A., STEGGLES P., WARD A., WEBSTER P., “The anatomy of a context-aware application”. In 5th ACM MOBICOM, August 1999.

[30] HAYKIN, Simon., "Neural Networks: a comprehensive foundation", Prentice Hall - 2nd Edition, 1999.

[31] HAZAS M., WARD A., “A novel broadband ultrasonic location system” In Fourth International Conference on Ubiquitous Computing, UbiComp 2002, 2002.

[32] HIGHTOWER J., BORRIELLO G., WANG R., “SpotON: An indoor 3D location sensing technology based on RF signal strength”, The Univ. of Washington, Technical Report: UW-CSE 2000-02-02, Feb 2000.

[33] HIGHTOWER J., VAKILI C., BORRIELLO G., WANG R., "Design and Calibration of the SpotON Ad-Hoc Location Sensing System", UW CSE 00-02-02, University of Washington, Department of Computer Science and Engineering, Seattle, WA, Feb. 2000.

[34] IEEE, IEEE Information Technology – “Telecommunications and Information Exchange between systems - Local and metropolitan area networks - Specific requirements - Parte 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications”, IEEE, 1999.

[35] IEEE., “IEEE Standard 802.11 - Wireless LAN medium access control (MAC) and physical layer (PHY) specifications”. 1999.

[36] IEEE, “IEEE Standard for IT-Telecommunications and information exchange between systems LAN/MAN - Part II:Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 4: Further Higher Data Rate Extension in the 2.4 GHz Band”, IEEE, 2003.

- [37] JI Y., "Location estimation in wireless networks", Ph.D Dissertation, Auburn University, 2005.
- [38] JOHNSON D. B., MALTZ D.A., "The Dynamic Source Routing Protocol in Ad Hoc Networks", Mobile Computing, 1996.
- [39] KOHONEN, T., "Self-Organizing Maps", Springer Series in Information Sciences, Springer, 1995.
- [40] KOHONEN, T., "The Self-Organizing Map", Proceedings of the IEEE, Vol.78, No.9, September, 1990.
- [41] KRISHNAN P., KRISHNAKUMAR A.S., JU W., MALLOWS C., GANU S., "A System for LEASE: System for location Estimation Assisted by Stationary Emitters for Indoor Wireless Networks", in Proceedings of IEEE Infocom 2004, Hong Kong, March, 2004.
- [42] KRUMM J., "Multi-camera multi-person tracking for easy living" In 3rd IEEE Int'l Workshop on Visual Surveillance, Piscataway, NJ, 2000.
- [43] LADD A. M., BEKRIS K., RUDYS A., MARCEAU G., KAVRAKI L. E., WALLACH D. S., "Robotics-based location sensing using wireless ethernet" In 8th ACM MOBICOM, Atlanta, GA, September 2002.
- [44] LEESE R., HURLEY S., "Methods and Algorithms for Radio Channel Assignment", Oxford University Press, London, 2002.
- [45] LINMARTZ J. M. G., "Wireless Communication, The Interactive Multimedia CD-ROM", University of Harvard - URL: [http://people.deas.harvard.edu/~jones/es151/prop\\_models/propagation.html](http://people.deas.harvard.edu/~jones/es151/prop_models/propagation.html), Acessado em Julho de 2009.
- [46] LOIOLA, Roberto R., *Aplicação de Redes Neurais Multicamadas para predição de campo elétrico em ambiente indoor*, Projeto Final em Engenharia Elétrica – 2006.



- [47] LOIOLA, Roberto R., ROMARIZ, Alexandre R.S., *Localização de dispositivos móveis em ambientes fechados utilizando redes neurais artificiais*, XII Encontro de Modelagem Computacional, Instituto Militar de Engenharia - IME, 2009.
- [48] LOIOLA, Roberto R., RAMOS, Francisco O.C., ROMARIZ, Alexandre R.S., *Sistemas Inteligentes - Uma aplicação à previsão da radio-propagação em ambientes fechados*, XI Encontro de Modelagem Computacional, Universidade Estadual do Rio de Janeiro, 2008.
- [49] MITCHELL, T.M., "Machine Learning". McGraw-Hill, New York - NY, USA, 1997.
- [50] MUKHERJEE A., BANDYOPADHYAY S., SAHA D., "Location Management and Routing in Mobile Wireless Networks", Artech House, Boston, 2003.
- [51] OHRTMAN F., ROEDER K., "Wi-Fi HandBook - Building 802.11b Wireless Networks", McGraw-Hill, Net York, 2003.
- [52] ORR R. J., ABOWD G. D., "The smart floor: A mechanism for natural user identification and tracking". In Conference on Human Factors in Computing Systems (CHI 2000), The Hague, Netherlands, April 2000.
- [53] PAHLAVAN K., KRISHNAMURTHY P., BENEAT J., "Wideband radio propagation modeling for indoor geolocation applications" IEEE Communications Magazine, April 1998.
- [54] PAHLAVAN K., LI X., MAKELA J. P., "Indoor geolocation science and technology", IEEE Communication Magazine, Vol. 40, No. 2, Feb.2002.
- [55] PAHLAVAN K., LI X., YLIANTTILA M., CHANA R.S., LATVA-AHO M., "An overview of wireless indoor geolocation techniques and systems" In Mobile and Wireless Communications Networks, May 2000.

- [56] PERES, S. M., Slides sobre C-Means e Fuzzy C-Means, URL: [www.ic.unicamp.br/~luciano/ACH2016/notasdeaula/10\\_kmeans.pdf](http://www.ic.unicamp.br/~luciano/ACH2016/notasdeaula/10_kmeans.pdf) .
- [57] PRIYANTHA N. B., CHAKRABORTY A., BALAKRISHNAN H., “The cricket location support system”. In 6th ACM MOBICOM, Boston, MA, August 2000.
- [58] ROMARIZ, A.R.S., *Apostila de Sistemas Inteligentes*, Setembro de 2005.
- [59] ROOS T., MYLLYMAKI P., TIRRI H., “A statistical modeling approach to location estimation” IEEE Transactions on Mobile Computing, January-March 2002.
- [60] ROOS T., MYLLYMAKI P., TIRRI H., MISIKANGAS P., SIEVANEN J., “A probabilistic approach to WLAN user location estimation” International Journal of Wireless Information Networks, July 2002.
- [61] SAHA S., CHAUDHURI K., SANGHI D., BHAGWAT P., “Location determination of a mobile device using IEEE 802.11b access point signals” In IEEE Wireless Communications and Networking Conference 2003, March 2003.
- [62] SAVARESE, C., “Robust Positioning Algorithms for Distributed Ad-Hoc Wireless Sensor Networks”, Master's Thesis, University of California, Berkeley, 2002.
- [63] SHREVE G., KELL D., “A precision location network using ultra wideband WLAN radios”, 2001.
- [64] SMAILAGIC A., SIEWIOREK D. P., ANHALT J., KOGAN D., WANG Y., “Location sensing and privacy in a context aware computing environment” Pervasive Computing, 2001.
- [65] STALLINGS, W., “Wireless Communications and Networks” Prentice Hall, first edition, 2002.

- [66] TAFNER, Malcon A., *Reconhecimento de palavras faladas isoladas usando redes neurais artificiais*, Dissertação de Mestrado em Engenharia Elétrica, Universidade Federal de Santa Catarina, 1996.
- [67] TAO P., RUDYS A., LADD A. M., WALLACH D. S., “Wireless LAN location-sensing for security applications” In Proceedings of the ACM Workshop on Wireless Security, San Diego, CA, 2003.
- [68] TEKINAY S., “Special issue on wireless geolocation systems and services” IEEE Communications Magazine, April 1998.
- [69] TERAOKA F.; GOGO K.; MITSUYA K.; SHIBUI R.; MITANI K., “Unified Layer 2 (LS) Abstractions - RFC 5184”, IETF - Internet Engineering Task Force, 2008.
- [70] WALLBAUM M., “Wheremops: An indoor geolocation system” In the 13th IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications, September 2002.
- [71] WANT R., HOPPER A., FALCO V., GIBBONS J., “The active badge location system” ACM Transactions on Information Systems, January 1992.
- [72] WANT R., SCHILIT B., “Expanding the horizons of location-aware computing”. IEEE Computer, August 2001.
- [73] WARD A., JONES A., HOPPER A., “A new location technique for the active office” IEEE Personal Communications, October 1997.
- [74] WASSI G.I., DESPINGS C., GRENIER D., NERGUIZIAN C., "Indoor location using received signal strength of IEEE 802.11b access point", Electrical and Computer Engineering, 2005, Canadian Conference on, 1-4 May 2005.
- [75] WERB J., LANZL C., “Designing a positioning system for finding things and people indoors” IEEE Spectrum, September 1998.

- [76] WIKIPEDIA, <http://pt.wikipedia.org/wiki/FLOPS>, Acessado em Julho de 2009.
- [77] WREN C. R., AZARBAYEJANI A., DARRELL T., PENTLAND A., “Pfinder: Real-time tracking of the human body” IEEE Transactions on Pattern Analysis and Machine Intelligence, 1997.
- [78] YOUSSEF M., AGRAWALA A., “Handling samples correlation in the horus system”.In IEEE Infocom 2004, 2004.
- [79] YOUSSEF M., AGRAWALA A., “On the optimality of WLAN location determination systems” Technical Report UMIACS-TR 2003-29 and CS-TR 4459, University of Maryland, College Park, March 2003.
- [80] YOUSSEF M., AGRAWALA A., “Small-scale compensation for WLAN location determination systems” In IEEE WCNC 2003, March 2003.
- [81] YOUSSEF M., AGRAWALA A., SHANKAR A. U., “WLAN location determination via clustering and probability distributions” In IEEE PerCom 2003, March 2003.
- [82] YOUSSEF M., AGRAWALA A., SHANKAR A. U., NOH S. H., “A probabilistic clustering-based indoor location determination system” Technical Report UMIACS-TR 2002-30 and CS-TR 4350, University of Maryland, College Park, March 2002.

## **APÊNDICES**

## **A – DESCRIÇÃO DOS APS UTILIZADOS NO EXPERIMENTO**

Nome: **WLPCI**

Localização: **Laboratório de Projeto de Circuito Integrado**

Andar: **Térreo**

Especificações técnicas:

Modelo: **DLINK High Speed 2,4 GHz**

Antenas: **Um dipolo externo**

Diagrama das Antenas: **Omnidirecional**

Frequência: **2,4 – 2,4835 GHz (De acordo com as normas do País)**

Throughput máximo (Velocidade): **54 MBps**

Potência Máxima: **18 dBm**

Canal de Operação: **01**

Segurança: **WPA-TKIP**

MAC Address: **00:15:e9:63:37:26**

Nome: **LabFontes**

Localização: **Laboratório de Pesquisa de Fontes Alternativas de Energia**

Andar: **Térreo**

Especificações técnicas:

Modelo: **3Com OfficeConnect® Wireless 108 Mbps 11g Cable/DSL Router (3CRWER200-75)**

Antenas: **Dois dipolos externos**

Diagrama das Antenas: **Omnidirecional**

Frequência: **2,4 – 2,4835 GHz (De acordo com as normas do País)**

Throughput máximo (Velocidade): **802.11g (108, 54, 48, 36, 24, 18, 12, 9, & 6 Mbps)**

Potência Máxima: **18 dBm**

Canal de Operação: **11**

Segurança: **RSNA-CCMP**

MAC Address: **00:1c:c5:d7:77:e0**

Nome: **GRAV**

Localização: **Laboratório de Pesquisa, Robótica e Automação**

Andar: **Sobreloja**

Especificações técnicas:

Modelo: **Edimax RS-100apg**

Antenas: **Um dipolo externo**

Diagrama das Antenas: **Omnidirecional**

Frequência: **2,4 – 2,4835 GHz (De acordo com as normas do País)**

Throughput máximo (Velocidade): **802.11g**

Potência Máxima: **19 dBm**

Canal de Operação: **11**

Segurança: **WPA-TKIP**

MAC Address: **00:1e:e5:6d:29:1f**

Nome: **LAVSI**

Localização: **Laboratório de Pesquisa, de Controle e Visão por Computador**

Andar: **Sobreloja**

Especificações técnicas:

Modelo: **TPLINK Td-Wr642g**

Antenas: **Um dipolo externo**

Diagrama das Antenas: **Omnidirecional**

Frequência: **2,4 – 2,4835 GHz (De acordo com as normas do País)**

Throughput máximo (Velocidade): **802.11g**

Potência Máxima: **15 dBm**

Canal de Operação: **1**

Segurança: **WPA-TKIP**

MAC Address: **00:21:29:b6:6e:15**