

**UNIVERSIDADE REGIONAL DE BLUMENAU  
CENTRO DE CIÊNCIAS EXATAS E NATURAIS  
CURSO DE CIÊNCIAS DA COMPUTAÇÃO**

(Bacharelado)

**PROTÓTIPO DE SOFTWARE PARA CONTROLE DE  
ACESSO DE FUNCIONÁRIOS UTILIZANDO REDES  
NEURAS PARA IDENTIFICAÇÃO DE IMPRESSÃO DIGITAL**

TRABALHO DE CONCLUSÃO DE CURSO SUBMETIDO À UNIVERSIDADE  
REGIONAL DE BLUMENAU PARA A OBTENÇÃO DOS CRÉDITOS NA  
DISCIPLINA COM NOME EQUIVALENTE NO CURSO DE CIÊNCIAS DA  
COMPUTAÇÃO — BACHARELADO

**ALEXSANDRA ZAPAROLI**

BLUMENAU, JULHO/2002

2002/1-07

# **PROTÓTIPO DE SOFTWARE PARA CONTROLE DE ACESSO DE FUNCIONÁRIOS UTILIZANDO REDES NEURAS PARA IDENTIFICAÇÃO DE IMPRESSÃO DIGITAL**

**ALEXSANDRA ZAPAROLI**

ESTE TRABALHO DE CONCLUSÃO DE CURSO, FOI JULGADO ADEQUADO  
PARA OBTENÇÃO DOS CRÉDITOS NA DISCIPLINA DE TRABALHO DE  
CONCLUSÃO DE CURSO OBRIGATÓRIA PARA OBTENÇÃO DO TÍTULO DE:

**BACHAREL EM CIÊNCIAS DA COMPUTAÇÃO**

---

Prof. Jomi Fred Hubner — Orientador na FURB

---

Prof. José Roque Voltolini da Silva — Coordenador do TCC

**BANCA EXAMINADORA**

---

Prof. Jomi Fred Hubner

---

Prof. Roberto Heinzle

---

Prof. Dr. Oscar Dalfovo

“Podemos achar que tudo o que a vida nos oferece amanhã é repetir o que fizemos ontem e hoje. Mas, se prestarmos atenção, vamos reparar que nenhum dia é igual ao outro. Cada manhã traz uma benção escondida. Uma benção que só serve para esse dia, e que não pode ser guardada ou reaproveitada. Se não usarmos esse milagre hoje, ele se perderá. Esse milagre está nos detalhes do cotidiano. É preciso viver cada minuto. Nos detalhes está a saída para nossas confusões, a alegria de nossos bons momentos, a pista certa para a decisão que precisa ser tomada. Não podemos deixar nunca que cada dia pareça igual ao anterior – porque todos os dias são diferentes.”

(Paulo Coelho)

## **DEDICATÓRIA**

ESTE TRABALHO DE CONCLUSÃO DE CURSO É DEDICADO À MINHA FAMÍLIA, PELO APOIO E INCENTIVO RECEBIDOS DURANTE OS ANOS DE GRADUAÇÃO.

# AGRADECIMENTOS

Agradeço aos professores do curso, que me proporcionaram conhecimento para desenvolver este trabalho, ao professor Roberto Heinzle, professor Oscar Dalfovo e especialmente ao professor Jomi Fred Hubner, por orientar-me no desenvolvimento do mesmo.

Aos meu namorado Lindomar Oliveria Santos, que me apoiou em vários momentos de dificuldade.

Aos meus pais e irmã que não me deixaram desistir de um sonho que faz parte de nossas vidas e que estiveram presente em todos os momentos necessários.

Aos meus colegas e amigos que me ajudaram e não me deixaram desanimar principalmente ao colega e amigo Leandro Augusto Frata Fernandes, por me ajudar em horas de insegurança.

# SUMÁRIO

LISTA DE FIGURAS .....	VIII
LISTA DE QUADROS .....	IX
RESUMO .....	X
ABSTRACT .....	XI
1 INTRODUÇÃO .....	1
1.1 OBJETIVOS DO TRABALHO .....	2
1.2 ESTRUTURA DO TRABALHO .....	2
2 FUNDAMENTAÇÃO TEÓRICA.....	3
2.1 DATILOSCOPIA .....	3
2.2 REDES NEURAIS .....	3
2.3 TÉCNICA DE RETROPROPAGAÇÃO DE ERRO .....	8
2.4 ALGUNS MODELOS DE REDES NEURAIS ARTIFICIAIS.....	10
3 DESENVOLVIMENTO DO PROTÓTIPO .....	12
3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO.....	12
3.2 ESPECIFICAÇÃO .....	12
3.2.1 LISTA DE EVENTOS .....	12
3.2.2 DIAGRAMA DE CONTEXTO .....	13
3.2.3 PROCESSO .....	16
3.3 IMPLEMENTAÇÃO .....	21
3.3.1 TÉCNICAS E FERRAMENTAS UTILIZADAS.....	21
3.3.2 ADAPTAÇÕES DO CÓDIGO DA REDE NEURAL PARA O PROTÓTIPO.....	27
3.3.3 IMPLEMENTAÇÃO DE CADASTROS E RELATÓRIOS.....	27
3.3.4 OPERACIONALIDADE DA IMPLEMENTAÇÃO.....	28

3.4 RESULTADOS E DISCUSSÃO .....	33
4 CONCLUSÕES .....	35
4.1 EXTENSÕES .....	35
REFERÊNCIAS BIBLIOGRÁFICAS .....	37

## LISTA DE FIGURAS

FIGURA 1 - NEURÔNIO BIOLÓGICO .....	4
FIGURA 2 - REDE DE COMPUTAÇÃO NEURAL .....	4
FIGURA 3 - REDE NEURAL PERCEPTRON .....	5
FIGURA 4 - UNIDADE DE PROCESSAMENTO DO PERCEPTRON .....	6
FIGURA 5 - FUNÇÃO DE TRANSFERÊNCIA .....	7
FIGURA 6 - DIAGRAMA DE CONTEXTO .....	14
FIGURA 7 - MER LÓGICO .....	14
FIGURA 8 - MER FÍSICO .....	16
FIGURA 9 - DFD PARTICIONADO .....	17
FIGURA 10 - FLUXO DE FUNCIONAMENTO DO SISTEMA .....	20
FIGURA 11 - IMPRESSÃO DIGITAL 40 X 40 PIXELS .....	21
FIGURA 12 - IMPRESSÃO DIGITAL 40 X 40 PIXELS AMPLIADA .....	22
FIGURA 13 - CÓDIGO PARA GERAR A IMAGEM EM BITS .....	23
FIGURA 14 - EXEMPLO DE ARQUIVO PADRÕES .....	23
FIGURA 15 - ARQUITETURA DA REDE NEURAL .....	24
FIGURA 16 - TELA DE TREINAMENTO DA RN .....	25
FIGURA 17 - EXEMPLO DO ARQUIVO DE SAIDAS.ASC .....	26
FIGURA 18 - PARTE DO CÓDIGO DO BOTÃO DE AUTENTICAÇÃO .....	28
FIGURA 19 - TELA DE AUTENTICAÇÃO DA IMPRESSÃO DIGITAL .....	29
FIGURA 20 - TELA DE CADASTRO DO FUNCIONÁRIO .....	29
FIGURA 21 - TELA DE CADASTRO DE FALTAS .....	30
FIGURA 22 - TELA DE CADASTRO DE JUSTIFICATIVAS .....	30
FIGURA 23 - TELA DE CADASTRO DE FUNÇÕES .....	31
FIGURA 24 - EXEMPLO DE RELATÓRIO DE FALTAS .....	32
FIGURA 25 - TELA DE RELATÓRIO DE HORAS EXTRAS .....	32
FIGURA 26 - EXEMPLO DE RELATÓRIO DE HORAS EXTRAS .....	33



## LISTA DE QUADROS

QUADRO 1 - UNIDADE SIGMA .....	7
QUADRO 2 - REGRA DE APRENDIZAGEM .....	8
QUADRO 3 - APLICAÇÕES DE REDES NEURAIIS .....	11
QUADRO 4 - LISTA DE EVENTOS .....	13
QUADRO 5 - TAREFAS EXECUTADAS NA INSERÇÃO DE UMA IMPRESSÃO DIGITAL .....	19

## **RESUMO**

O presente trabalho descreve o desenvolvimento de um protótipo de software com a função de controlar o acesso de funcionários da Divisão de Administração do Campus da FURB utilizando a impressão digital para a identificação dos funcionários. Este protótipo visa um melhor controle sobre os horários de entrada e saída e automatizar algumas tarefas como relatórios de faltas e relatórios de quantidades de horas extras.

## **ABSTRACT**

The present work describes the development of an software prototipe which controls the employees access to the Divisão de Administração do Campus of the FURB being used the fingerprint for the employee identification. This prototipe aims a better control on the entrance and exit schedules and the automation of some tasks as reports of lacks and reports of amounts of overtime.

# 1 INTRODUÇÃO

A busca de um processo cada vez mais eficiente de identificação de indivíduos, vem de tempos remotos, sempre tentando melhorar, aperfeiçoar e facilitar a identificação de pessoas que pretendem entrar em determinados lugares. Antigamente as empresas tinham um número menor de funcionários e era controlável, mas o número de funcionários cresceu e a dificuldade apareceu, não havia como controlar os horários dos funcionários. Aí começaram os Livros Pontos, onde os funcionários assinavam seus nomes e horários de entrada e saída, mas ainda havia o obstáculo do analfabetismo, muitos que não sabiam escrever. Logo após surgiu o cartão batido através de relógio ponto e com a tecnologia foram desenvolvidos outros tipos de identificação, como exemplos o código de barra, cartão magnéticos e muitos outros, mas a maioria é burláveis.

A deficiência de controlar o acesso de pessoas ou funcionários em determinados lugares gera algumas situações em que só a confiança não basta, tanto a empresa quanto os funcionários perdem. Ao verificar esta dificuldade na Divisão de Administração do Campus da FURB, deparou-se com alguns problemas sérios de funcionários que chegam atrasados, outros que saem antes do horário final do expediente, pensando em uma maneira de controlar essas entradas e saídas, resolveu-se desenvolver um protótipo de software que faça esse controle automático, revelando assim o cumprimento ou não dos horários estabelecidos em lei. Tendo que, há uma grande ocorrência de horas extras e que todo o cálculo é feito na mão, porque não automatizar esta tarefa também, fazendo com que o sistema analise os horários e forneça relatórios automáticos que são enviados ao Departamento de Recursos Humanos para o posterior pagamento das mesmas.

No contexto apresentado acima, um dos problemas é como identificar o funcionário de forma correta, sem que ele engane o sistema. Muitas análises foram feitas, atualmente na Universidade Regional de Blumenau, deveria usar o crachá com código de barra, mas não é feito com sucesso, pois qualquer pessoa poderia registrar a entrada ou saída, sem ser o usuário do crachá.

A justificativa é a automatização de processos que hoje são feitos manualmente. O que se espera com a conclusão deste trabalho é a certeza de que os funcionários terão seus horários monitorados, não podendo enganar seus empregadores e uma das conseqüências da

automatização é a agilidade de relatórios que os responsáveis pelo controle terão em suas mãos.

## **1.1 OBJETIVOS DO TRABALHO**

O objetivo principal deste trabalho é desenvolver um protótipo de software que automatize a função de controle de acesso de funcionários da Divisão de Administração do Campus, facilitando a disponibilidade de relatórios necessários para o controle de entrada e saída dos funcionários. Não havendo mais o problema de faltas e atrasos freqüentes sem o conhecimento de seus empregadores.

Uma das maneiras de resolver este problema seria o registro da impressão digital, que é única e intransferível para qualquer indivíduo. Uma maneira prática e eficiente de controlar os horários de funcionários que de uma forma ou outra conseguem enganar seus empregadores. Será usado RN para treinar e reconhecer as impressões digitais Sem falar em outras utilidades que se pode desenvolver com a impressão digital.

## **1.2 ESTRUTURA DO TRABALHO**

A seguir serão descritos brevemente cada capítulo do trabalho.

O segundo capítulo apresenta uma fundamentação teórica das áreas envolvidas, que são: Datiloscopia, Redes Neurais especificamente o modelo perceptron e a técnica de retropropagação de erro. Foi escolhido RN pela facilidade que a rede tem de receber informações e compará-las com outras informações em um tempo muito curto.

No terceiro capítulo é apresentado o desenvolvimento do trabalho, comentando as técnicas e as ferramentas envolvidas. Uma das ferramentas utilizada é o Power Designer que facilitará na especificação das fases do projeto, e outra ferramenta que será utilizada será o ambiente Delphi.

No capítulo quatro será apresentado a implementação, como acontece o treinamento da rede neural e a operacionalidade da implementação.

Por fim, no quinto capítulo são apresentadas as principais conclusões e sugestões para futuros trabalhos.

## 2 FUNDAMENTAÇÃO TEÓRICA

As seções seguintes abordam os assuntos necessários para o desenvolvimento do trabalho.

### 2.1 DATILOSCOPIA

Datilosopia é o processo de identificação humana por meio das impressões digitais. Identidade, em um conceito mais amplo, é a identificação única e imutável dos indivíduos.

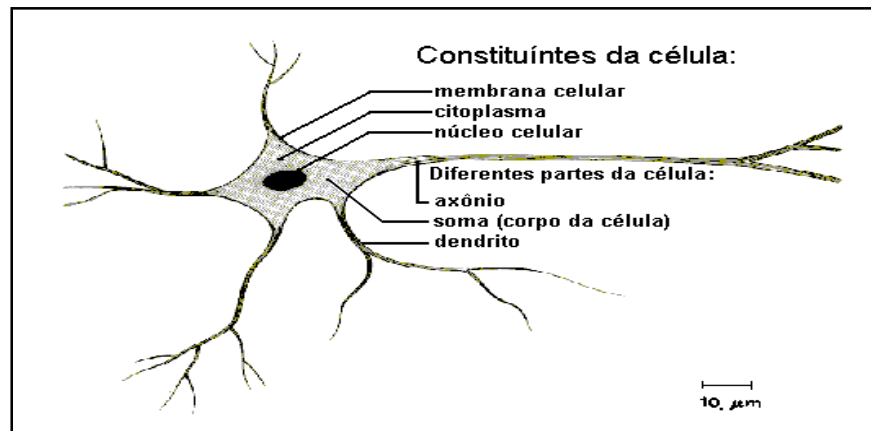
A datilosopia divide-se em civil, criminal e clínica. A datilosopia civil trata da identificação para a cédula de identidade, carteira profissional etc. A datilosopia criminal tem três finalidades, a identificação do inquirido policial, a expedição de documentos de idoneidade e a identificação de fragmentos de impressões digitais, coletadas em locais de crimes. A datilosopia clínica é a que cuida das perturbações que se verificam nos desenhos digitais, em consequência de algumas doenças ou do exercício de certas profissões (Tavares Jr.,1991).

Muitas técnicas são utilizadas para o reconhecimento de uma impressão digital, podemos destacar as minúcias que são pontos inimitáveis das digitais. Geralmente, uma minúcia é a terminação de uma *crista* ou um *vale* das digitais. A posição relativa das minúcias é um atributo inimitável de uma pessoa. Até mesmo os gêmeos possuem minúcias diferentes e, portanto, digitais diferentes. As impressões digitais completas consistem de mais de cem minúcias.

### 2.2 REDES NEURAIS

O cérebro humano possui cerca de 10 bilhões de neurônios biológicos. São as células mais diferenciadas do organismo, pois apresentam a maior complexidade estrutural e funcional, um exemplo de neurônio biológico é apresentado na fig. 1.

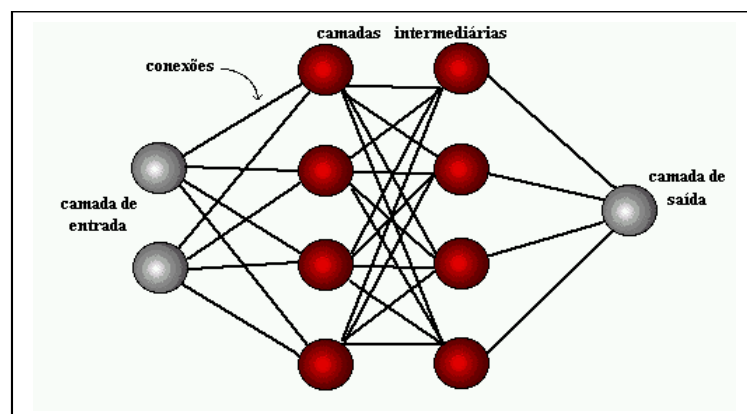
FIGURA 1 - NEURÔNIO BIOLÓGICO



FONTE: Silva (1999)

Redes Neurais Artificiais são sistemas computacionais, de implementação em hardware ou software, que imitam as habilidades computacionais do sistema nervoso biológico, usando um grande número de simples neurônios artificiais interconectados. Os neurônios artificiais são emulações simplificadas dos neurônios biológicos, estes recebem informação de sensores ou de outros neurônios artificiais, produzindo operações simples sobre estes dados, e passam o resultado para outros neurônios artificiais (Loesch, 1996). Uma RN é uma coleção de neurônios dispostos de forma que configurem um aspecto específico, como a fig. 2. É com estes neurônios que a RN aprenderá as informações que serão fornecidas pelos canais de entrada dos neurônios. O aprendizado está distribuído por toda a rede, ou seja, por todos os neurônios (Tafner, 1995).

FIGURA 2 - REDE DE COMPUTAÇÃO NEURAL



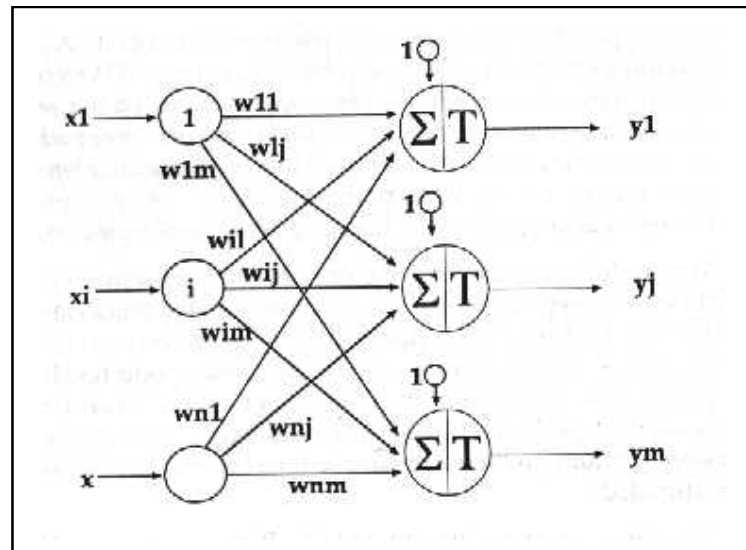
FONTE: Silva (1999)

O modelo de RN perceptron foi o primeiro a ser desenvolvido. Possui um conceito de neurônio artificial, exemplificado na fig. 4, que ainda hoje é usado. Cada neurônio computa uma soma ponderada de suas entradas, e passa esta soma em uma função não-linear com limiarização. O algoritmo de aprendizagem para o perceptron permite distinguir classes no conjunto de entradas, se estas fossem linearmente separáveis em termos de algum espaço de decisão. Perceptrons tomam decisões, determinam se um padrão de entrada se encaixa ou não em um certo padrão (Loesch, 1996).

A RN perceptron é formada pela camada de entrada, onde cada elemento da camada de entrada faz a distribuição do sinal que ele recebe para todas as unidades de processamento. Os neurônios que não são de entrada são constituídos de unidades sigma ( $\Sigma$ ) e de funções de transferência (T) (Tafner, 1995).

A RN perceptron, ilustrada na fig. 3, consiste de um conjunto de unidades sensoriais (nós de fonte) que constituem a camada de entrada, uma ou mais camadas ocultas de nós computacionais e uma camada de saída de nós computacionais. O sinal de entrada se propaga para frente através da rede, camada por camada.

FIGURA 3 - REDE NEURAL PERCEPTRON

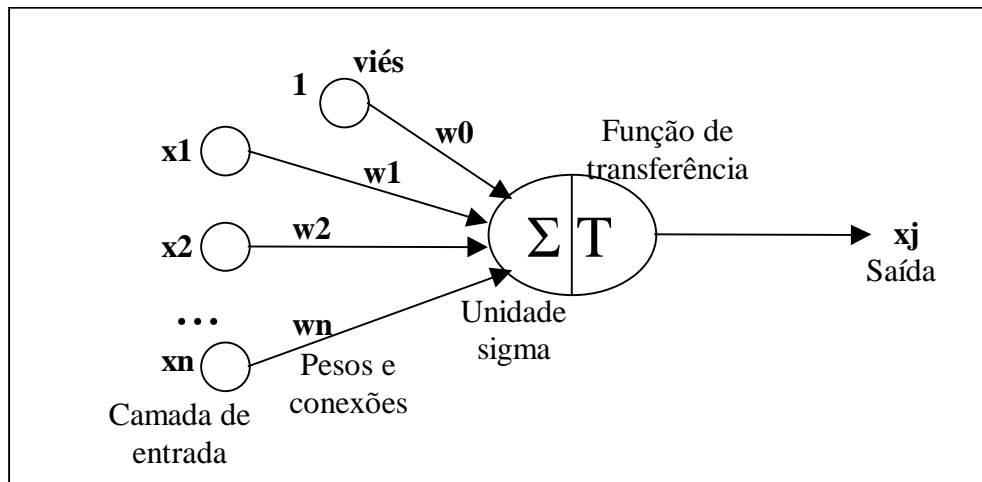


FONTE: Tafner (1995)

O neurônio artificial do perceptron ou a unidade de processamento é ser representada na fig. 4.



FIGURA 4 - UNIDADE DE PROCESSAMENTO DO PERCEPTRON



FONTE: Tafner (1995)

a) Camadas

- **Camada de Entrada:** O processamento dá-se em valores numéricos, onde em alguns casos cada valor de entrada corresponde a um único atributo. O pré-processamento faz-se necessário quando do tratamento de problemas envolvendo atributos qualitativos ou imagens, a fim de convertê-los para valores numéricos equivalentes (Turban, 1993).
- **Camada de Saída:** Nesta camada encontra-se a solução para o problema proposto. Assim como a camada de entrada não foge a regra, podendo somente assumir valores numéricos.

b) Conexões

- **Peso das Conexões:** Os pesos são elementos fundamentais a rede. Eles exprimem a força relativa em valor numérico, dos dados introduzidos inicialmente ou às várias conexões que transferem dados de uma camada para outra.
- **Unidade Sigma:** Efetua os cálculos de multiplicação e soma a fim de estabelecer a média dos pesos que compreendem as conexões dos neurônios da camada de entrada.
- **Função de Transferência:** Com base no resultado da função soma, o neurônio produz ou não determinada saída. A partir de então a função de transferência modifica o valor de saída do neurônio para que o mesmo respeite uma determinada faixa de domínio.

Em uma unidade de processamento do perceptron, tem-se várias etapas até se chegar ao resultado esperado. O viés é um elemento da camada de entrada, cujo sinal tem sempre valor 1. A camada de entrada apenas conduz o sinal que chega ao perceptron até sua unidade sigma, representada pela quadro 1. Os pesos estão associados às conexões que levam os sinais vindos da camada de entrada até a unidade sigma e que simulam as eficácias das sinapses do neurônio biológico. A unidade sigma faz a soma ponderada dos sinais de entrada, ou seja, multiplica cada sinal de entrada pelo peso correspondente à conexão por onde passa o sinal, fazendo a soma total destes produtos.

QUADRO 1 - UNIDADE SIGMA

$$Y_j = \sum_{i=0}^n X_i \cdot W_{ij}$$

FONTE: Tafner (1995)

As variáveis da equação da unidade sigma são descritas a seguir:

- a)  $Y_j$  = saída da rede neural do elemento j;
- b)  $\sum$  = somatório dos valores de entradas multiplicados por seus pesos;
- c)  $x_i$  = valor de entrada do índice i;
- d)  $w_{ij}$  = peso associado ao elemento j de índice i.

E por fim, a função de transferência, fig. 5, que determina a saída do perceptron conforme o valor que lhe for passado pela unidade sigma.

FIGURA 5 - FUNÇÃO DE TRANSFERÊNCIA

$$\begin{aligned} T &= \{ \Sigma > 0 \quad Y_j = 1 \\ T &= \{ \Sigma \leq 0 \quad Y_j = 0 \end{aligned}$$

FONTE: Tafner (1995)

O perceptron suporta um aprendizado supervisionado, o que significa que para treinar a rede um conjunto de dados de treinamento deve ser construído e padrões de treinamento

devem ser definidos. No caso do perceptron, é modificado cada peso (conexão) de acordo com a quadro 2.

QUADRO 2 - REGRA DE APRENDIZAGEM

$$W_{ij} := W_{ij} + \mu \delta_j \cdot X_i$$

**FONTE: Tafner (1995)**

As variáveis da equação da regra de aprendizagem são descritas a seguir:

- a)  $W_{ij}$  = peso associado ao elemento j de índice i;
- b)  $\delta_j$  = valor do erro;
- c)  $\mu$  = taxa de aprendizado;
- d)  $X_i$  = valor de entrada do índice i.

O algoritmo de treinamento do perceptron simples, uma camada apenas, obedece aos seguintes passos:

- a) 1º Passo: arbitrar pesos para as conexões entre a camada de entrada e a unidade sigma;
- b) 2º Passo: aplicar um padrão de sinais de entrada  $x_j$  e calcular a soma ponderada S;
- c) 3º Passo: passar a soma ponderada S para a função de transferência:
  - se a saída  $y_j$  estiver correta, voltar ao 2º passo;
  - se a saída  $y_j$  estiver errada e for 0, adicionar cada peso das conexões com os sinais de entrada relativos a elas;
  - se a saída  $y_j$  estiver errada e for 1, subtrair de cada peso das conexões os sinais de entrada relativos a elas;
- d) 4º Passo: voltar ao 2º passo.

## 2.3 TÉCNICA DE RETROPROPAGAÇÃO DE ERRO

Os perceptrons são aplicados com sucesso para resolver diversos problemas difíceis, através do seu treinamento, existem perceptrons de camada única e perceptrons de multi

camadas um deles muito conhecido, o algoritmo de retropropagação de erro (*error back-propagation*) (Haykin, 2001).

O termo de retropropagação de erro refere-se a um método de treinamento empregado para os perceptrons de multi camadas, no qual os pesos das conexões são ajustados através da chamada regra delta generalizada. A rede pode ser treinada para acompanhar uma larga variedade de mapeamentos, alguns dos quais muitos complexos. Isso porque os elementos de processamento da camada oculta aprendem a responder a características encontradas na entrada. Estas características referem-se a correlações de atividade entre diferentes nodos de entrada, possibilitando uma representação abstrata da informação de entrada na camada oculta. Além da capacidade de abstração, a rede possui capacidade de generalização, ou seja, é capaz de classificar corretamente um padrão complexo mesmo quando este não pertencer ao conjunto de treinamento da rede. Por exemplo: se uma rede aprendeu a distinguir linhas côncavas de convexas, através de exemplos, e conseguiu abstrair este conceito, pode-se apresentar uma nova linha para a rede e ela poderá ser capaz de classifica-la corretamente, desde que tenha conseguido abstrair o conceito de concavidade (Loesch, 1993).

A aprendizagem por retropropagação de erro consiste de dois passos através das diferentes camadas da rede: um passo para frente, a propagação, e um passo para trás, a retropropagação. No passo para frente, um padrão de atividade (vetor de entrada) 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. Durante o passo de propagação, os pesos sinápticos da rede são todos fixos. Durante o passo para trás, os pesos sinápticos são todos ajustados de acordo com uma regra de correção de erro. Especificamente, 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. 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 (Haykin, 2001).

O algoritmo de retropropagação de erro obedece aos seguintes passos:

- a) 1º Passo: inicialize os pesos  $w_{ij}^{(k)}$  da rede com valores aleatórios próximos a zero (isto assegura que a rede não está saturada por valores grandes na rede, e previne contra outras patologias de treinamento, como por exemplo, se todos os pesos fossem

inicializados por igual, e o vetor de saída desejada contivesse componente desigual, a rede não aprenderia);

b) 2º Passo: Use o par de treinamento  $(x,d)$ . Aplique o vetor  $x$  na camada de entrada e o vetor da resposta desejada  $d$  e propague os sinais da rede até a camada de saída. Seja  $y$  a saída da rede. Calcule o erro quadrático  $\mathcal{E}^2 = \sum_{j=1}^m (d_j - y_j)^2$ . Se for inferior ao valor de tolerância, pare com sucesso. Senão prossiga ao passo 3;

c) 3º Passo: Faça  $k =$  última camada;

d) 4º Passo: Para todo elemento  $j$  da camada  $K$  faça:

- Calcule  $\mathcal{E}_j^{(k)}$  empregando  $\mathcal{E}_j^{(k)} = d_j - x_j^{(k)} = d_j - y_j$  se  $k$  for a última camada;

ou  $\mathcal{E}_j^{(k)} = \sum_{i=1}^{N_{k+1}} (\delta_i^{(k+1)} \cdot w_{ij}^{(k+1)})$  se for uma camada oculta;

- Calcule  $\delta_j^{(k)}$  empregando  $\delta_j^{(k)} = \mathcal{E}_j^{(k)} \cdot f' (S_j^{(k)})$ ;

e) 5º Passo:  $k \leftarrow k-1$ . Se  $k > 0$  vá para o passo 4, senão prossiga no passo 6;

f) 6º Passo: Recalcule todos os pesos de conexão da rede empregando

$w_j^{(k)}(n+1) = w_j^{(k)}(n) + 2\mu \delta_j^{(k)}(n) X_j^{(k)}(n)$ , tome outro par de treinamento e retorne ao passo 2.

A maioria absoluta das aplicações existentes compõe-se de redes neurais com aprendizado supervisionado, que pode ser considerado como a capacidade que a rede possui de modificar o seu desempenho a partir da comparação entre a resposta obtida e a resposta desejada. A partir das entradas fornecidas, os pesos das conexões são ajustados por cálculo até obter-se as saídas desejadas.

## 2.4 ALGUNS MODELOS DE REDES NEURAIS ARTIFICIAIS

O quadro 3 resume os modelos das RN, as aplicações, as vantagens e desvantagens, foi escolhido o modelo perceptron de multi camada e a técnica de retropropagação de erro(back-propagation) devido à vantagem da operação ser mais rápida, bem compreendida e com muitas aplicações de sucesso.

QUADRO 3 - APLICAÇÕES DE REDES NEURAIIS

Modelo	Aplicações Básicas	Ano	Vantagens	Desvantagens
Adaline/Madaline	Filtragem de sinal adaptativo, equalização adaptativa	1960	Rápida e Fácil de implementar	Somente é possível classificar espaços linearmente separáveis
Adaptative Resonance Theory (ART)	Reconhecimento de Padrões	1983	Capaz de aprender novos padrões	Naturaza dos exemplares categóricos podem mudar com o aprendizado
Backpropagation Perceptron	Reconhecimento de padrões, filtragem de sinal, controle robótico, compressão de dados, segmentação de sinal, etc.	1974-1986	Operação rápida, bem compreendida e com muitas aplicações de sucesso	Tempo de treinamento longo
BAM – Memória Associativa Bidirecional	Heteroassociativa (memória endereçada por conteúdo)	1987	Simples, prova clara da estabilidade dinâmica	Capacidade de armazenamento e precisão recuperação pobres
Boltzmann Machine, Cauchy Machine	Reconhecimento de padrões (imagens, sons, radar), otimização	1984	Capaz de formar representação ótima das características dos padrões	Tempo de aprendizado longo
Brain-State-in-a-Box (BSB)	Revocação autoassociativa	1977	Bom desempenho	Incompleta em termos de aplicações em potencial
Hopfield	Evocação autoassociativa, otimização	1982	Simples e estabilidade dinâmica	Incapaz de aprender novos estados
Neocognitron	Reconhecimento de caracteres manuscritos / imagens	1975	Capaz de reconhecer padrões independente da escala, translação e rotação	Estrutura complexa
Quantização de Vetor de aprendizagem	Revocação autoassociativa (complementação de uma padrão a outro parcial apresentado), compressão de dados	1981	Capaz de auto-organizar vetores	Treinamento lento
Recurrent	Controle robótico, reconhecimento de fala, previsão do elemento seqüencial	1987	Excelente para classificação, mapeamento de informações variando no tempo	Rede complexa
Redes de ligações funcionais	Classificação, mapeamento	1988	Rápida para treinar	Não é claro o modo de identificar funções adotadas
Time-Delay	Reconhecimento de fala	1987	Rápida operação	Janela fixada para a atividade temporal representada

FONTE: Loesch (1996)

## 3 DESENVOLVIMENTO DO PROTÓTIPO

De acordo com o que foi discutido na introdução, e apresentado como objetivo deste trabalho, foi desenvolvido um protótipo de software para controle de acesso de funcionários da Divisão de Administração do Campus da FURB, utilizando redes neurais para a identificação da impressão digital. O presente capítulo trata da especificação e implementação deste protótipo.

### 3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

O usuário quer cadastros simples de funcionários, de funções, de setores, de locais, se seções, de faltas e justificativas, apenas com as informações necessárias. Relatórios de faltas e de horas extras simples e curtos. E principalmente, o usuário precisa de um sistema que controle os horários dos funcionários, possibilitando a verificação imediata da existência de horas extras.

### 3.2 ESPECIFICAÇÃO

O propósito deste protótipo é controlar a entrada e saída (ponto) de funcionários utilizando a impressão digital como chave para o processo de identificação. Sendo integrado ao sistema, a disponibilidade de relatórios de horas extras e de faltas.

Foi definida também a análise estruturada, para melhor compreensão do problema e para organização o do protótipo.

#### 3.2.1 LISTA DE EVENTOS

A lista de eventos está definida no quadro 4, indicando como procede a análise estruturada. Abaixo está relacionado o que cada evento faz.

- a) **Responsável cadastra Funções:** o responsável cadastra as funções dos funcionários da Divisão de Administração do Campus;
- b) **Responsável cadastra Seções:** o responsável cadastra a seção em que o funcionário da Divisão de Administração do Campus pertence;

- c) **Responsável cadastra Setores:** o responsável cadastra o setor em que o funcionário da Divisão de Administração do Campus está lotado;
- d) **Responsável cadastra Locais:** o responsável cadastra o local em que o funcionário da Divisão de Administração do Campus trabalha;
- e) **Responsável cadastra Licenças:** o responsável cadastra as licenças que os funcionários tem direito;
- f) **Responsável cadastra as Faltas:** o responsável cadastra os dias em que o funcionário faltou;
- g) **Responsável cadastra as Justificativas:** o responsável cadastra as justificativas que o funcionário apresenta;
- h) **Funcionário é Cadastrado:** Os dados do funcionário são cadastrados;
- i) **Funcionário cadastra Horário:** Funcionário registra o horário com sua impressão digital;
- j) **Gerar relatório de Faltas:** o responsável gera o relatório de faltas para cadastrar as mesmas e se necessário descontar na folha de pagamento;
- k) **Gerar relatório de Horas Extras:** o responsável gera o relatório de horas extras para enviar ao DRH para devidos pagamentos.

#### QUADRO 4 - LISTA DE EVENTOS

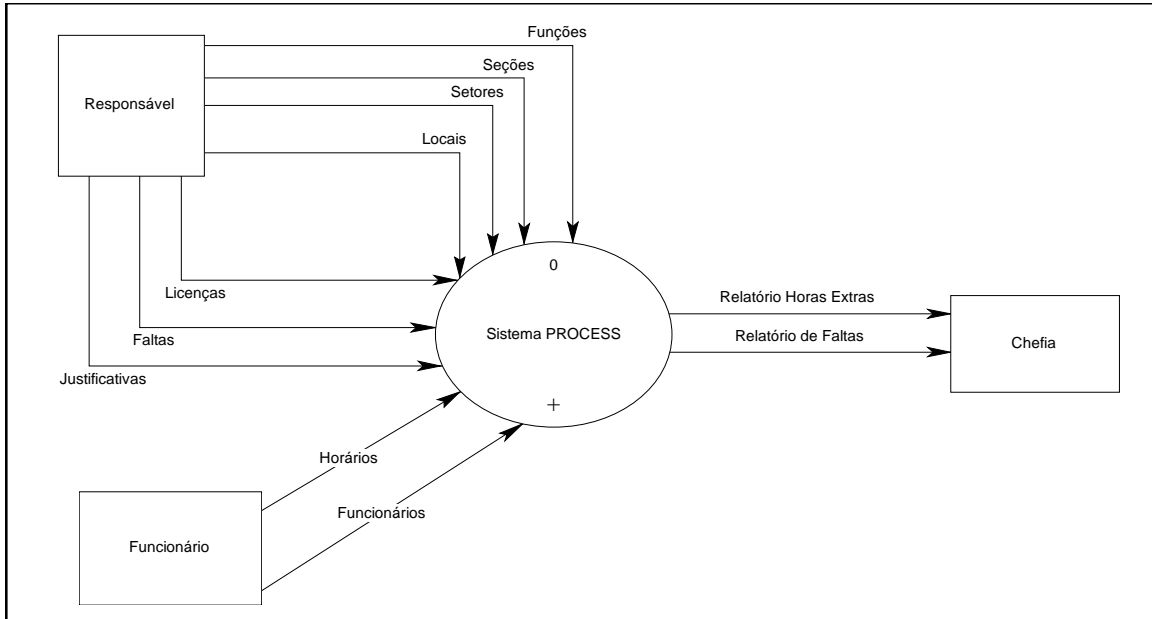
1 – Responsável Cadastra Funções
2 - Responsável Cadastra Seções
3 - Responsável Cadastra Setores
4 - Responsável Cadastra Locais
5 - Responsável Cadastra Licenças
6 - Responsável Cadastra Faltas
7 - Responsável Cadastra Justificativas
8 – Funcionário Cadastra Horário
9 - Funcionário é Cadastrado
10 – Gerar Relatório de Faltas
11 – Gerar Relatório de Horas Extras

### 3.2.2 DIAGRAMA DE CONTEXTO

O Diagrama de Contexto representa o sistema por um único processo e suas iterações com entidades externas, são definidos na fig. 6.

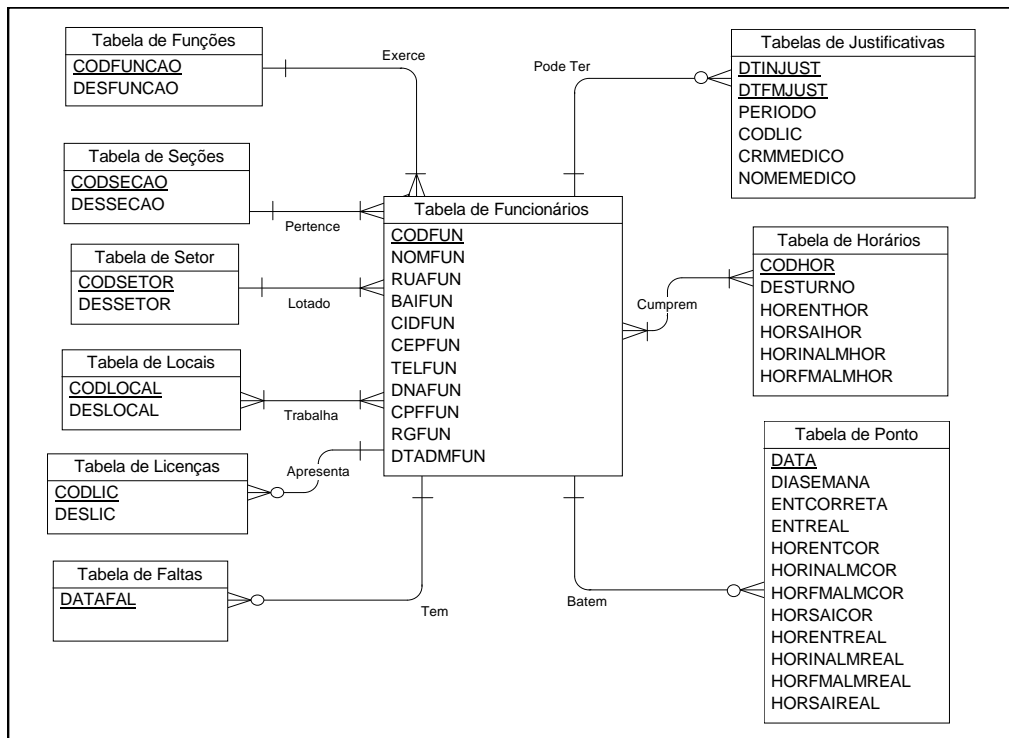


FIGURA 6 - DIAGRAMA DE CONTEXTO



Na fig. 7 está definido o MER Lógico que indica todas as variáveis utilizadas para identificação dos funcionários.

FIGURA 7 - MER LÓGICO



A tabela de Funções, Seções, Setores, Locais e Licenças contém o código e a descrição dos itens.

A tabela de Faltas contém os dias em que o funcionário faltou, possibilitando a impressão de relatórios com a mesma.

A tabela de Justificativas contém as datas em que a justificativa está valendo, o período, o motivo (licença) e o CRM (Conselho Regional de Medicina) e o nome do médico no caso de atestados.

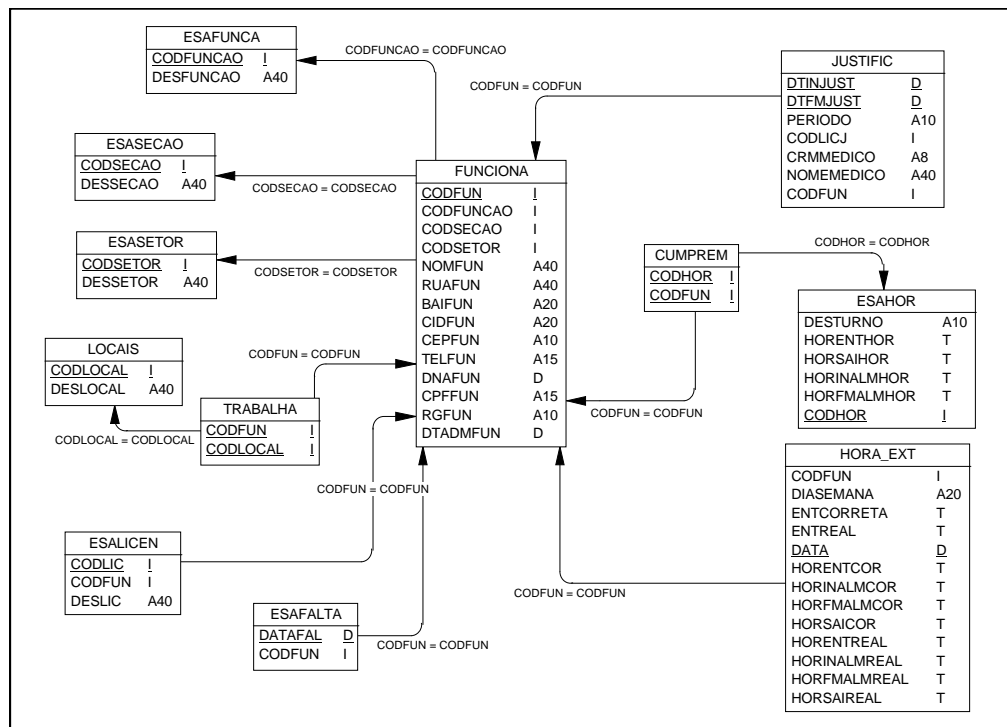
A tabela de Funcionários contém os dados gerais dos funcionários comuns a qualquer empresa que tenha um cadastro de funcionários.

A tabela de Horários contém os horários corretos que cada funcionário deve cumprir.

A tabela de Ponto contém os horários reais que o funcionário cumpriu em sua jornada de trabalho, possibilitando a impressão de relatórios com a mesma.

A partir do MER Lógico, é gerado o MER Físico que está representado na fig. 8, que indica quais as ligações entre as entidades.

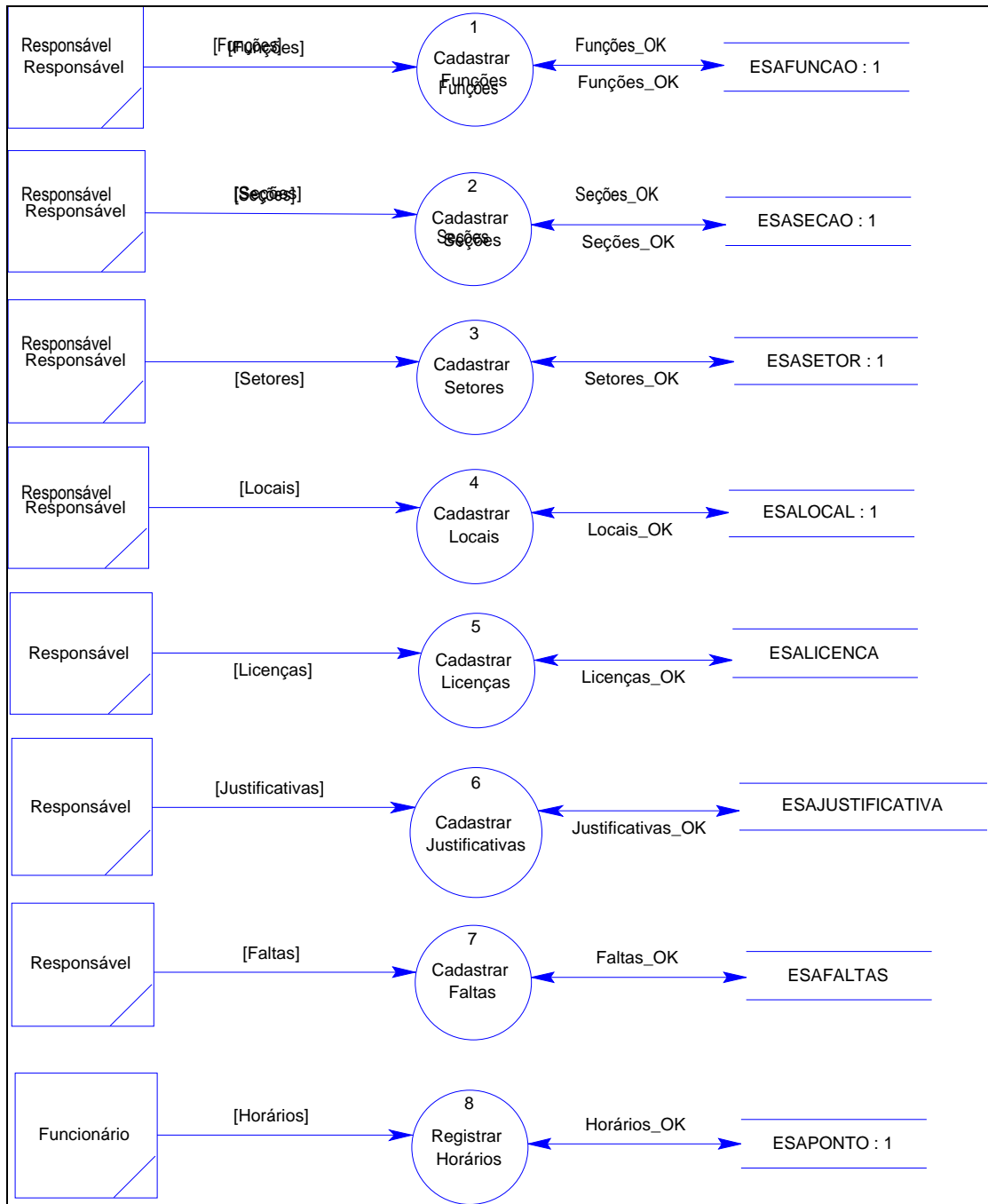
FIGURA 8 - MER FÍSICO

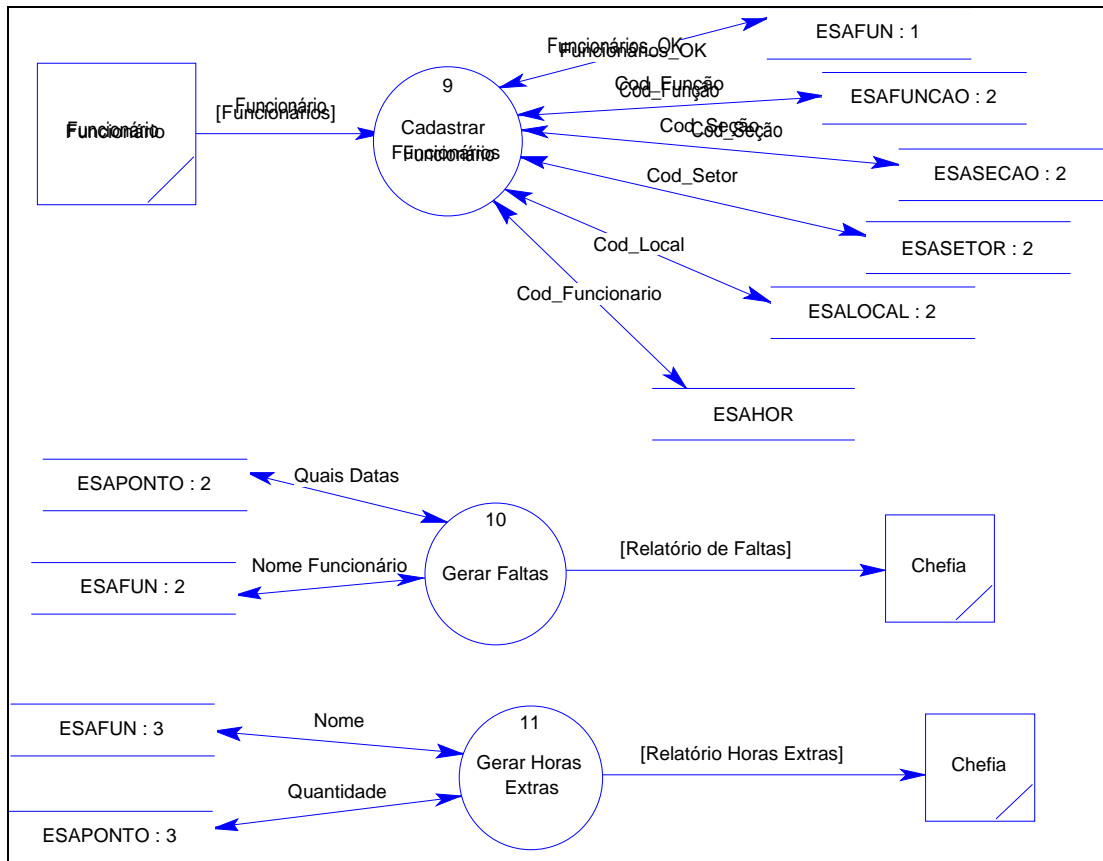


### 3.2.3 PROCESSO

A partir do Diagrama de Contexto é gerado o DFD Particionado, ilustrado na fig. 9 também conhecido como DFD de resposta aos eventos. Para cada evento, deve-se representar os processos, as entidades externas, os depósitos de dados, os fluxos e respostas, na forma de um DFD.

FIGURA 9 - DFD PARTICIONADO





O detalhamento de cada entidade está descrito abaixo, o que cada entidade faz e qual a sua importância.

- 1) **Cadastrar Função:** Recebe informações do responsável, verifica se já existe a função cadastrada, se não existir grava, se existir abandona.
- 2) **Cadastrar Seção:** Recebe informações do responsável, verifica se já existe a seção cadastrada, se não existir grava, se existir abandona.
- 3) **Cadastrar Setor:** Recebe informações do responsável, verifica se já existe o setor cadastrado, se não existir grava, se existir abandona.
- 4) **Cadastrar Local:** Recebe informações do responsável, verifica se já existe o local cadastrado, se não existir grava, se existir abandona.
- 5) **Cadastrar Licenças:** Recebe informações do responsável, verifica se já existe o motivo da licença cadastrado, se não existir grava, se existir abandona.
- 6) **Cadastrar Justificativa:** Recebe informações do responsável, verifica se já existe a justificativa cadastrada, se não existir grava, se existir abandona.

- 7) **Cadastrar Faltas:** Recebe informações do responsável, verifica se já existe a falta cadastrada correspondente ao dia de determinado funcionário, se não existir grava, se existir abandona.
- 8) **Registrar Horários:** Funcionário digitaliza a impressão digital, o sistema compara as impressões do banco de dados, se localizou a digital, verifica a data e o horário em tempo real se já existem armazenados no banco de dados, se não existirem grava, se existir abandona.
- 9) **Cadastrar Funcionário:** Recebe as informações de funcionário, seleciona a função do funcionário, a seção que o funcionário pertence, o setor que o funcionário está lotado, o local que o funcionário trabalha e o horário que ele cumpre, se não existir grava, se existir abandona.
- 10) **Gerar Faltas:** a chefia digita um período de tempo e gera um relatório, que contém uma listagem de todos os funcionários que faltaram e as respectivas datas de faltas.
- 11) **Gerar Horas Extras:** a chefia digita um período de tempo e gera um relatório, que contém todos os funcionários que tem um somatório de horas além do seu horário de trabalho normal.

Destes processos, somente a entidade (8) será especificado em detalhes. As demais tem um funcionamento muito simples.

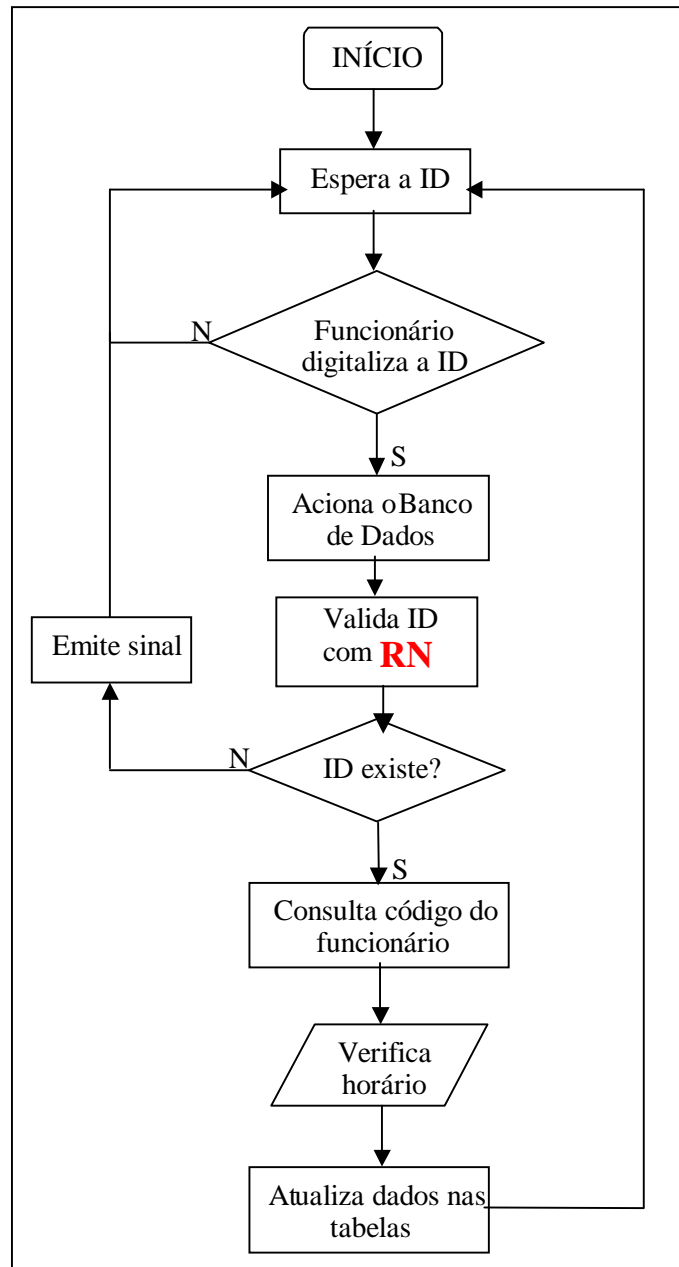
O quadro 5 mostra numa linguagem descritiva das tarefas que o sistema deve executar na entidade (8), ao receber uma impressão digital, por ser a tarefa mais complexa.

#### QUADRO 5 - TAREFAS EXECUTADAS NA INSERÇÃO DE UMA IMPRESSÃO DIGITAL

- Sistema espera a impressão digital;
- Sistema recebe a impressão digital e aciona o banco de dados para fazer a comparação;
- Sistema compara a impressão digital recebida com as impressões digitais armazenadas no banco de dados;
- Se a impressão existir no banco de dados, o sistema envia os dados necessários do funcionário correspondente à impressão digital para as tabelas;
- Se a impressão não existir no banco de dados, o sistema emite um sinal, e não recebe nenhum dado referente ao funcionário.

De acordo com as tarefas executadas pelo sistema ao receber uma impressão digital, foi definido o fluxo de funcionamento da rotina de gerenciamento do sistema, conforme mostra a fig. 10.

FIGURA 10 - FLUXO DE FUNCIONAMENTO DO SISTEMA



### 3.3 IMPLEMENTAÇÃO

O sistema utiliza apenas as informações previamente cadastradas e combinada com a digitalização da impressão digital, verifica se o funcionário existe, a resposta do sistema é dada com o preenchimento dos dados do funcionário, como os horários em tempo real em que ele está digitalizando a impressão digital. Com isso quando o responsável pelo controle dos horários precisar de relatórios de faltas ou de horas extras, basta acessar o sistema e imprimir o relatório.

#### 3.3.1 TÉCNICAS E FERRAMENTAS UTILIZADAS

Baseando-se nas pesquisas realizadas nos TCC's de Silva (1999) e Valdameri (1997) e em resultados obtidos com os mesmos, optou-se pela utilização dos componentes de modelo de rede perceptron com aprendizado de retropropagação de erro, que Valdameri (1997) desenvolveu em seu TCC. A escolha deste modelo de RN, deu-se principalmente pela grande capacidade de generalização e na sua rápida operacionalização.

##### 3.3.1.1 PRÉ-PROCESSAMENTO DAS IMAGENS

A imagem não pode ser identificada pela RN como nós identificamos a olho nu. A RN deve receber um arquivo com informações em bit, ou seja, 0's e 1's, por isso é necessário um pré-processamento da imagem.

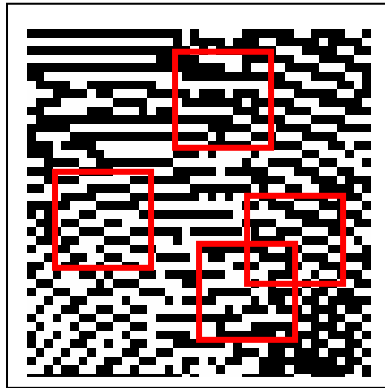
FIGURA 11 - IMPRESSÃO DIGITAL 40 X 40 PIXELS



As dimensões da imagem da impressão digital são 40 x 40 pixels, representada na fig. 11. Se fosse considerado esta imagem, sem diminuições ou filtros, a quantidade de neurônios seria de 1600 x 2 na camada de entrada. Isto causaria um tempo de treinamento muito grande. Por esta razão a imagem será reduzida gerando 4 blocos de 10 X 10 pixels cada, totalizando 400 pixels ou 400 neurônios. Na fig. 13, pode-se ver o código de como são obtidos os 400 neurônios. A figura é em preto e branco, 1 assume a cor preta e 0 assume a cor branca, e é contado cada pixel e armazenado em um arquivo chamado Padrões.asc. Na fig. 12 está exemplificado quais as partes da imagem da impressão digital que são transformados em blocos de 10 pixels cada.



FIGURA 12 - IMPRESSÃO DIGITAL 40 X 40 PIXELS AMPLIADA



A autenticação da digital deveria ser feita através de um leitor óptico, que reconheceria a digital do funcionário e informaria os dados do mesmo ao sistema. Mas tendo em vista que o leitor óptico custar caro, está sendo usado imagens de impressões digitais *escaneadas* para fazer o treinamento e os testes para o protótipo.

FIGURA 13 - CÓDIGO PARA GERAR A IMAGEM EM BITS

```

function ImagemParaTexto(ImagemEntrada: TImage): String;
function ObterBloco(inlin, fmlin, incol, fncol: Integer): String;
var
  contlin,
  contcol : integer;
begin
  result := "";
  for contlin := inlin to fmlin do begin
    for contcol := incol to fncol do begin
      if ImagemEntrada.Canvas.Pixels [contLin, ContCol] = clblack then
        result := result + '1'
      else
        result := result + '0';
    end;
  end;
end;
begin
  result := ObterBloco(6, 15, 16, 25) +
    ObterBloco(16, 25, 6, 15) +
    ObterBloco(21, 30, 26, 35) +
    ObterBloco(26, 35, 21, 30);
end;

```

Na Fig. 14 está um exemplo da geração de uma imagem bmp em um arquivo txt (Padrões.asc).

FIGURA 14 - EXEMPLO DE ARQUIVO PADRÕES

```

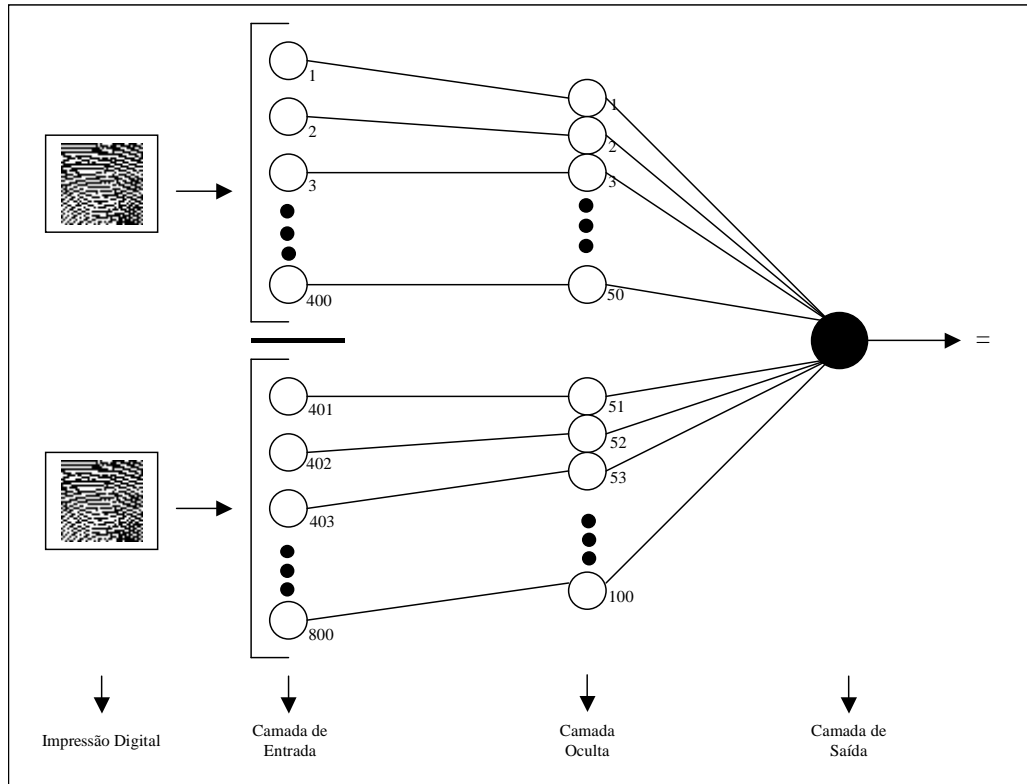
* Padrões submetidos a treinamento, exemplo de 1 imagem
i 1 1 1 0 0 1 0 1 0 1 1 0 0 1 0 0 1 1 1 1 1 0 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 0 1 0 1 1 0 1 1 0 0 1 1 1 1 1 1 0 0 1 1 1 1 1 0 0 1 1 1 1 1 0 1 1 1 0 1
1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 0 1 1 0 1 1 1 1 1 1 0 1 0 1 0 1 0 1 0 1 1 1 1
0 1 0 0 0 1 0 1 0 1 0 1 1 0 0 1 0 1 1 1 0 1 1 0 1 1 0 1 0 0 1 0 1 0 1 0 1 0 1 0 1
1 1 1 1 1 1 0 1 0 1 0 1 0 0 1 1 0 1 0 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 0 1 0 1 0 1 1 0
1 0 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1
1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1
1 0 0 1 0 1 1 1 1 1 1 0 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0
1 1 0 1 1 0 0 1 1 1 1 1 1 0 0 1 1 0 1 1 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 0
1 1 1 1 1 0 1 1 1 0 1 1 0 1 1 1 1 1 1 1 0 1 0 1 0 1 0 1 1 1 0 1 0 0 0 1 0 1 0 1 0
1 0 1 1 0 0 1 0 1 1 1 0 1 1 0 1 0 1 0 0 1 0 1 0 1 1 0 1 0 1 1 1 1 1 1 1 1 1 0 1 0
1 0 0 1 1 0 1 0 1 0 1 1 1 0 1 1 1 0 1 0 1 0 1 0 1 0 1 1 1 0 1 0 1 0 1 0 1 0 1 1 1
1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
d 0.2

```

\* Obs: O símbolo “\*” representa uma linha de comentário  
 O caracter “i” no início da linha identifica como sendo um padrão de entrada  
 O caracter “d” no início da linha identifica como sendo um padrão de saída desejado

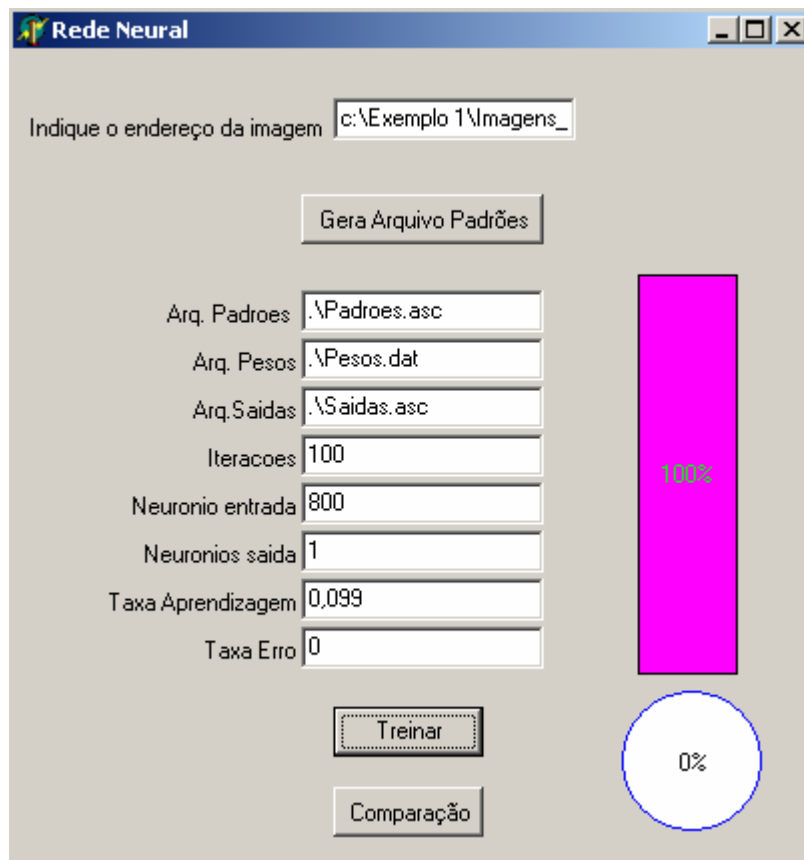
### 3.3.1.2 REDE NEURAL

FIGURA 15 - ARQUITETURA DA REDE NEURAL



- a) **Camada de Entrada:** contém 800 neurônios, 400 neurônios são de 1 impressão do leitor óptico e 400 neurônios são de outra impressão do banco de dados, podendo ser a mesma impressão ou não.
- b) **Camada Oculta:** recebe o número de neurônios da camada de entrada e divide por 8, o resultado será o número da camada oculta.
- c) **Camada de Saída:** O neurônio da camada de saída será a resposta da RN, a resposta pode ser de igualdade se as impressões forem iguais ou desigualdade se as impressões forem diferentes.

FIGURA 16 - TELA DE TREINAMENTO DA RN



Os parâmetros utilizados para treinar a RN são:

- a) **Arq. Padrões:** contém os padrões gerados a partir da imagem pré-processada;
- b) **Arq. Pesos:** é gerado no treinamento da RN;
- c) **Arq. Saída:** é gerado no término do treinamento da RN;
- d) **Iterações:** contém o número de iterações para o treinamento da rede;
- e) **Neurônio Entrada:** contém o numero de neurônios da camada de entrada;
- f) **Neurônio Saída:** conte o número do neurônios da camada de saída;
- g) **Taxa Aprendizagem:** contém a taxa de aprendizagem com a qual o modelo da RN será treinado;
- h) **Taxa Erro:** contém a taxa de erro tolerável que o modelo da RN poderá suportar.

Para a RN aprender ela precisa ser treinada, neste protótipo foram usados 20 exemplos de imagens *escaneadas* para treinar a rede. Foram usados exemplos iguais e diferentes, para a rede poder aprender. Assim que a rede termina de treinar, pode ser feita a comparação, onde a

RN cria um arquivo chamado Saídas.asc com o resultado da comparação. Um exemplo de Saídas.asc está na Fig. 17.

FIGURA 17 - EXEMPLO DO ARQUIVO DE SAIDAS.ASC

```
* Resultado da revocação da dos Padrões Apresentados - Exemplo 1
r 0.80
* Resultado da revocação da dos Padrões Apresentados - Exemplo 2
r 0.80
* Resultado da revocação da dos Padrões Apresentados - Exemplo 3
r 0.80
* Resultado da revocação da dos Padrões Apresentados - Exemplo 4
r 0.80
* Resultado da revocação da dos Padrões Apresentados - Exemplo 5
r 0.80
* Resultado da revocação da dos Padrões Apresentados - Exemplo 6
r 0.80
* Resultado da revocação da dos Padrões Apresentados - Exemplo 7
r 0.80
* Resultado da revocação da dos Padrões Apresentados - Exemplo 8
r 0.80
* Resultado da revocação da dos Padrões Apresentados - Exemplo 9
r 0.80
* Resultado da revocação da dos Padrões Apresentados - Exemplo 10
r 0.80
```

\* Obs: O símbolo “\*” representa uma linha de comentário

O caracter “r” no início da linha identifica como sendo uma saída do modelo

Mais detalhes sobre o código da RN utilizado podem ser encontrados em Valdameri (1997), que desenvolveu um protótipo utilizando RN aplicadas ao sistema de informação do jogo de empresas virtual. Neste protótipo ele desenvolveu os componentes de RN para utilização na ferramenta Delphi que foram reutilizadas neste trabalho.

### 3.3.1.3 POWER DESIGNER

A ferramenta Power Designer possibilita a análise estruturada de uma aplicação, fornecendo assim os passos e um estudo antecipado dos itens que serão necessários para a implementação do protótipo.

#### 3.3.1.4 DELPHI

A ferramenta de programação visual utilizada para o desenvolvimento do protótipo foi o Delphi, pois possui características indispensáveis para elaboração de aplicações facilitando principalmente a geração de cadastros.

Os dados que são cadastrados, são armazenados nas tabelas do Paradox, um tipo de banco de dados compatível com o Delphi.

#### 3.3.2 ADAPTAÇÕES DO CÓDIGO DA REDE NEURAL PARA O PROTÓTIPO

Foram feitas algumas alterações no código da implementação da RN que Valdameri (1997) desenvolveu, como o tamanho das variáveis, de *byte* passou a *word*, algumas variáveis *string* passaram a ser *ansistring*. Essas alterações foram feitas por causa da quantidade de caracteres que a variável aceita, por exemplo, *string* aceitava apenas 255 caracteres de entrada e o arquivo de padrões exige 800 caracteres.

A quantidade máxima de neurônios para a rede era declarada em uma variável global e aceitava apenas 20 neurônios, foi alterada a quantidade para 800 neurônios. A quantidade de neurônios na camada oculta recebeu novo divisor, anteriormente a camada oculta recebia o número de camada de entrada e dividia por 2, deixando a camada oculta com 400 neurônios, isso à torna muito grande e lenta, agora ela é dividida por 8 deixando a camada oculta com apenas 100 neurônios.

#### 3.3.3 IMPLEMENTAÇÃO DE CADASTROS E RELATÓRIOS

O protótipo possui cadastros que foram desenvolvidos utilizando os componentes do Delphi, eles já têm incluídas grande parte do código necessário para a implementação das funções de cadastro.

Grande parte da implementação foi dispensada para fazer os relatórios que envolvem cálculos que utilizam informações de várias tabelas. A Fig. 18 descreve uma parte da implementação do botão de Autenticação que é ilustrado na fig. 18 (principal tela).

FIGURA 18 – PARTE DO CÓDIGO DO BOTÃO DE AUTENTICAÇÃO

```

// procurar funcionário
Encontrou := False;
dmDados.QryAutenticacao.Open;
dmDados.QryAutenticacao.First;
while (not Encontrou) and (not dmDados.QryAutenticacao.Eof) do
begin
  ArquivoBMPBase := dmDados.BMPFunCodigo
                    (dmDados.QryAutenticacaoCODFUN.AsInteger);
  if FileExists(ArquivoBMPBase) then
  begin
    ImageBancoDados.Picture.LoadFromFile(ArquivoBMPBase);
    TxtBase := ImagemParaTexto(ImageBancoDados);
    ArquivoParaComparacao(TxtBase, TxtNovo, '0.2', Revoca.ArquivoPadres);

    SetCurrentDir(ExtractFileDir(Application.ExeName));

    Revoca.AbrirRede;
    Revoca.Revocar;
    Saida := PegarSaida(Revoca.ArquivoSaida);

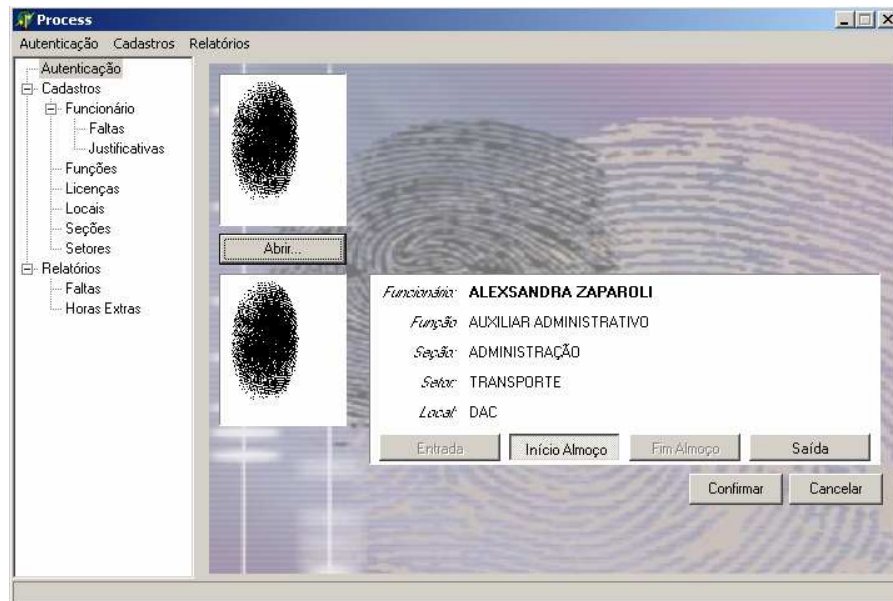
    if Saida <= 0.21 then
      Encontrou := True
    else
      dmDados.QryAutenticacao.Next;
    end
  else
    dmDados.QryAutenticacao.Next;
  end;
end;

```

### 3.3.4 OPERACIONALIDADE DA IMPLEMENTAÇÃO

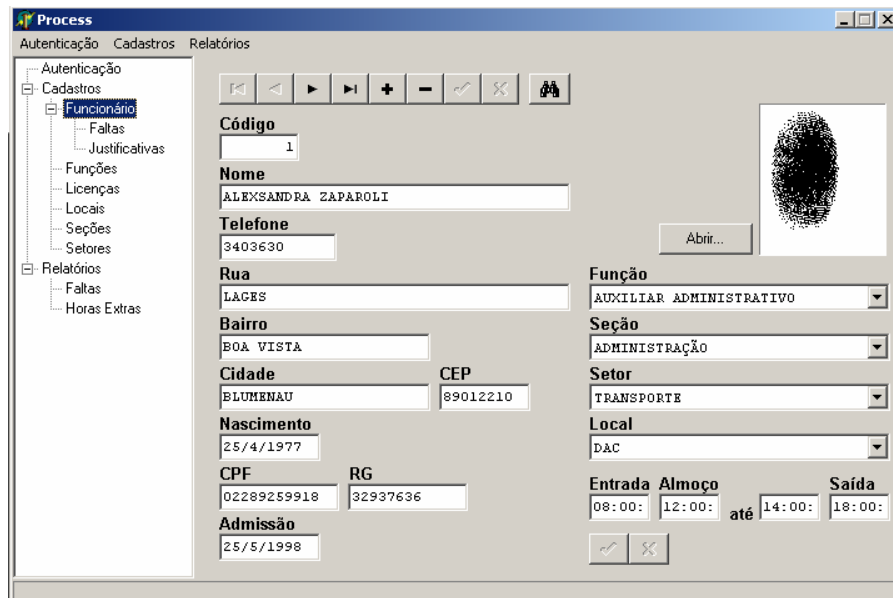
A fig. 19 representa a tela de autenticação do funcionário, é nela que o usuário digitaliza sua impressão digital e assume uma indicação e Entrada, Início Almoço, Fim Almoço ou Saída, todos os horários que forem diferenciados do horário correto do funcionário é considerado hora extra, com a prévia autorização do responsável.

FIGURA 19 - TELA DE AUTENTICAÇÃO DA IMPRESSÃO DIGITAL



Na fig. 20, o responsável cadastra os dados gerais do funcionário, podendo relacionar os dados com a imagem da impressão digital.

FIGURA 20 - TELA DE CADASTRO DO FUNCIONÁRIO



A tela de cadastro de faltas é representada pela fig. 21. Quando o funcionário não digitaliza a impressão digital nos horários corretos e por um motivo ou outro ele falta, é seu dever trazer uma justificativa. Para esse dia não ser descontado em sua folha de pagamento o



responsável precisa cadastrar os dias que o funcionário faltou e se ele trouxer a justificativa, é só preencher com os dados na tela de cadastro de justificativas.

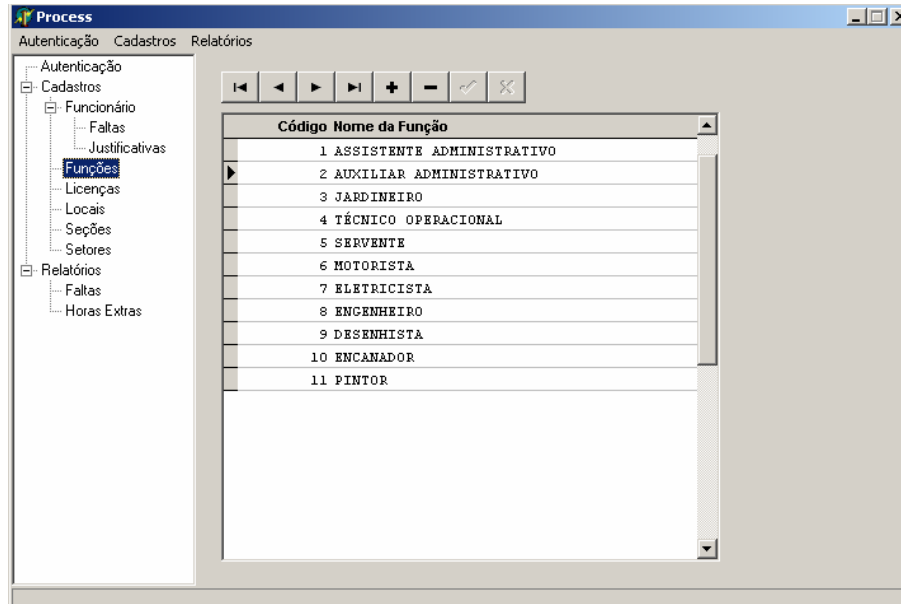
FIGURA 21 - TELA DE CADASTRO DE FALTAS

A fig. 22 representa a tela de cadastro das justificativas. Nesta tela são cadastrados os diversos atestados ou dispensas que podem ser aceitas pelos responsáveis no controle dos horários.

FIGURA 22 - TELA DE CADASTRO DE JUSTIFICATIVAS

O cadastro de Funções é representado na fig. 23, as demais telas de cadastros de Locais, Licenças, Seções e Setores são idênticas a de Funções, pois tem os mesmos requisitos como código e descrição.

FIGURA 23 - TELA DE CADASTRO DE FUNÇÕES



As telas de relatórios de faltas e de horas extras são idênticas e simples também, o responsável apenas indica qual o período ele quer, e clica no botão Visualizar, onde são gerados os relatórios com as faltas e com horas extras feita por todos os funcionários. Na Fig. 24 está um exemplo do relatório de faltas.

Figura 24 - EXEMPLO DE RELATÓRIO DE FALTAS

<b>Relatório de Faltas</b>		16/07/02 20:01 Página 1
<b>Funcionário:</b> DAMA CENO FAUSTO	<b>Função</b> ELETRICISTA <b>Setor:</b> ELÉTRICO <b>Local:</b> DAC	
Faltas: 23/04/02, 26/05/02, 14/06/02, 10/07/02.		
<b>Funcionário:</b> FLAVIA CARVALHO	<b>Função</b> AUXILIAR ADMINISTRATIVO <b>Setor:</b> TRANSPORTE <b>Local:</b> DAC	
Faltas: 15/04/02, 17/05/02, 23/05/02, 28/06/02, 29/06/02.		
<b>Funcionário:</b> JONATHAN NAZÁRIO	<b>Função</b> AUXILIAR ADMINISTRATIVO <b>Setor:</b> SERVIÇOS GERAIS <b>Local:</b> DAC	
Faltas: 15/06/02.		

Normalmente os relatórios são emitidos no final do mês, visando obter os horários em que os funcionários trabalharam.

FIGURA 25 - TELA DE RELATÓRIO DE HORAS EXTRAS

Na Fig. 26 podemos ver um exemplo simples de relatório de horas extras.

FIGURA 26 - EXEMPLO DE RELATÓRIO DE HORAS EXTRAS

<b>Relatório de Horas Extras</b>			
16/07/02 20:41 Página 1			
<hr/>			
Funcionário: ALEXSANDRA ZAPAROLI	Data	Horários	
	01/07/02	07:00:00	às 08:00:00
	02/07/02	12:00:00	às 14:00:00
	02/07/02	18:00:00	às 19:00:00
	15/07/02	12:00:00	às 14:59:53
<hr/>			
Funcionário: AMANDO OTTERSBACK	Data	Horários	às
<hr/>			
Funcionário: CARLOS ROBERTO LINHARES	Data	Horários	
	01/06/02	07:00:00	às 08:00:00
	01/06/02	12:00:00	às 13:00:00
	05/07/02	13:00:00	às 14:00:00
	05/07/02	18:00:00	às 19:00:00
	15/07/02	00:15:25	às 08:00:00
	15/07/02	12:00:00	às 14:00:00

### 3.4 RESULTADOS E DISCUSSÃO

No desenvolvimento deste protótipo, foram utilizadas algumas ferramentas, o Power Designer, ajudou para a análise do problema e quais indicações para resolução do mesmo.

O Delphi proporcionou uma interface amigável ao usuário, e é muito útil por já ter incluso grande parte dos códigos de implementação.

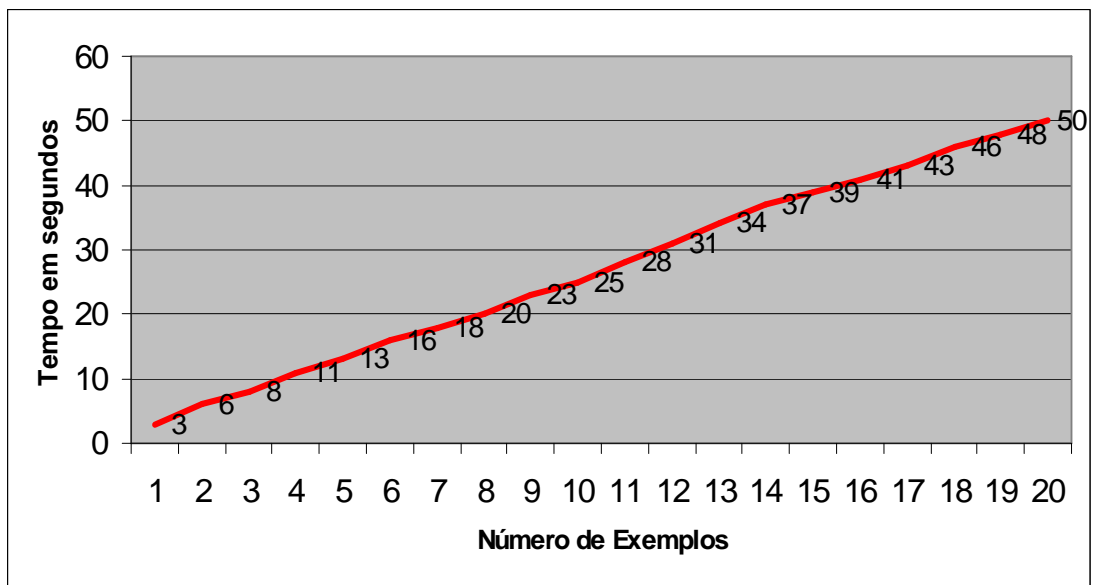
Os componentes que Valdameri (1997) desenvolveu foram utilizados para treinar os exemplos das impressões digitais.

Resultados do primeiro treinamento de 20 exemplos de impressões digitais:

- a) **Tempo:** a RN levou 50 segundos para treinar 20 exemplos mistos (iguais e diferentes) de impressões digitais, utilizando a numeração de 100 iterações;
- b) **Treino:** a RN acertou 100% dos 20 exemplos submetidos em treinamento;
- c) **Teste:** foram submetidos 10 novos exemplos para comparação, a RN acertou 100%.

Com os resultados gerados com o treinamento das impressões, foi feito um gráfico que objetiva a análise de quanto tempo leva cada exemplo para treinar.

GRÁFICO 1 - RESULTADO DO TEMPO DE TREINAMENTO



Após mais alguns testes, verificou-se que a identificação das impressões digitais foi de apenas 70%. O ideal neste caso seria o treinamento da rede com um número maior de impressões.

## 4 CONCLUSÕES

O objetivo do trabalho foi parcialmente alcançado. Os cadastros e relatórios estão de acordo com os requisitos levantados. Contudo não foi possível testar o conjunto do sistema com o leitor óptico.

As tecnologias utilizadas foram de grande ajuda. A análise estruturada auxiliou na pré-modelagem do protótipo, mas para uma próxima modelagem seria interessante utilizar a análise orientada a objeto. O ambiente Delphi foi utilizado por ter uma interface amigável e por que a RN já havia sido desenvolvida em Delphi.

No início do desenvolvimento do protótipo tentou-se utilizar a RN do TCC que Silva (1999) desenvolveu, mas não foi possível pois ele apenas classificava e não diferenciava a impressão digital. Mais tarde tentou-se desenvolver a RN, mas a dificuldade foi grande, pois levaria muito tempo para desenvolver e treinar. Por fim utilizei a RN que Valdameri (1997) desenvolveu, precisando apenas fazer alguns ajustes e treinar as impressões digitais.

Em alguns dos testes realizados a RN aprendeu 100% das impressões submetidas, contudo em algumas circunstâncias a RN não acertou 100% dos casos. Poderia-se fazer novas verificações, com um número maior de impressões de treinamento, para ver se o resultado da RN será de 100%.

O resultado obtido com a simulação da impressão como se fosse o leitor óptico atende a todas as necessidades.

Se o protótipo fosse utilizado pelo Departamento de Recursos Humanos, teria que ser adaptado para outros tipos de relatórios que eventualmente o Departamento deve utilizar. Se fosse utilizado para atender 1000 funcionários, acho que funcionaria, precisando apenas de treinamento com um número maior de impressões.

### 4.1 EXTENSÕES

Sugere-se o estudo de novos modelos de redes neurais a fim de atender com maior eficácia as necessidades propostas inicialmente neste trabalho, assim como sua implementação.

Sugere-se ainda o treinamento com um número maior e impressões digitais e com impressões não legíveis.

Existem ainda várias áreas em que a tomada de decisões seria de um grande apoio, várias áreas em que as redes neurais poderão auxiliar de várias formas.

## REFERÊNCIAS BIBLIOGRÁFICAS

ENGEL, Paulo Matirns. Desenvolvimento de um módulo neural para tarefas de reconhecimento de padrões. In: Simpósio Brasileiro de Inteligência Artificial, 10., 1993, Porto Alegre. **Anais...** Porto Alegre: UFRGS, 1993. p. 445-457.

GONZALEZ, Rafael C.; WOODS, Richard E. **Processamento de imagens digitais**. São Paulo: Edgard Blücher Ltda, 2000.

HAYKIN, Simon. **Redes neurais: princípios e prática**. Porto Alegre: Bookman, 2001.

LOESCH, Cláudio. Back-propagation para redes neuronais. **Dynamis Tecno-Científica**, Blumenau, v. 1, n. 4, p. 59-78, jul./set. 1993.

LOESCH, Cláudio; SARI, Solange Teresinha. **Redes neurais artificiais: fundamentos e modelos**. Blumenau: Editora da FURB, 1996.

SILVA, Alex Sandro da. **Protótipo de software para classificação de impressão digital**. 1999. 70 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

SOUZA, André Luiz. **Criação de tabelas utilizando o Sybase Central**, Uberaba, [2000?]. Disponível em: < <http://xfk.vilabol.uol.com.br/any/sqlany.htm> >. Acesso em: 03 jun. 2002.

TAVARES Jr, Gilberto da Silva. **A papiloscopia nos locais de crime: manual prático e teórico**. São Paulo: Ícone, 1991.

TAFNER, Malcon Anderson. **Redes neurais artificiais: introdução e princípio de neurocomputação**. Blumenau: Editora da FURB, 1995.

TURBAN, Efraim. **Decision support and expert systems**. New York: Macmillan, 1993.

VALDAMERI, Alexander R. **Redes neurais aplicadas ao sistema de informações do jogo de empresas virtual**. 1997. 69 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.