

# **Protótipo de um Sistema de Identificação do(s) delta(s) e núcleo em Impressões Digitais utilizando Redes Neurais Artificiais**

---

Acadêmico: Caio Rafael Silva Matias

Orientador: Prof. Jomi Fred Hübner

# Roteiro da Apresentação

- ✓ Introdução
- ✓ Fundamentação Teórica
- ✓ Desenvolvimento
- ✓ Conclusões

# Introdução

- ✓ Identificação de indivíduos (3 tipos)
- ✓ Identificação através das impressões digitais
  - Delta(s)
  - Núcleo

## Objetivos

Desenvolver um protótipo de identificação de indivíduos através de suas impressões digitais utilizando para isto pontos característicos destas, sendo eles os delta(s) e núcleo

# Fundamentação Teórica

- ✓ Impressões Digitais
- ✓ Biometria
- ✓ Processamento de Imagens
- ✓ Redes Neurais

# Impressões Digitais

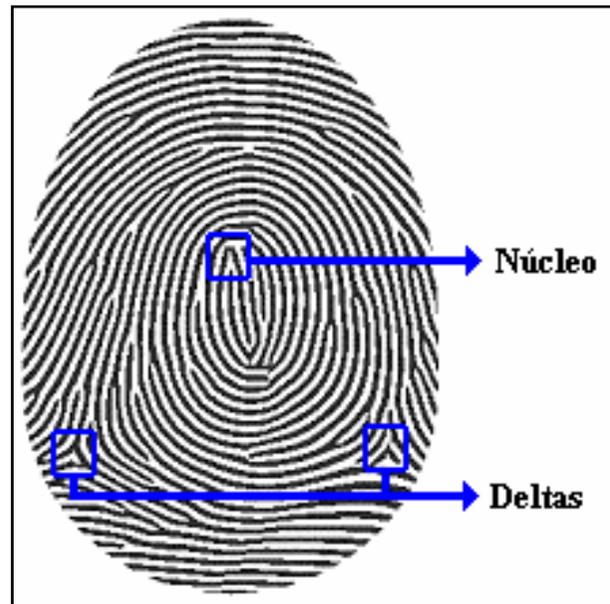
- ✓ Desenho digital e impressão digital.
- ✓ Estudo das impressões digitais: pré-histórico, empírico, científico (1664, 1891).
- ✓ Período científico definiu a Papiloscopia:
  - Datiloscopia
  - Quiroscopia
  - Poroscopia

# Impressões Digitais

- ✓ Postulados da Datiloscopia:
  - Perenidade
  - Imutabilidade
  - Variabilidade
  - Classificabilidade

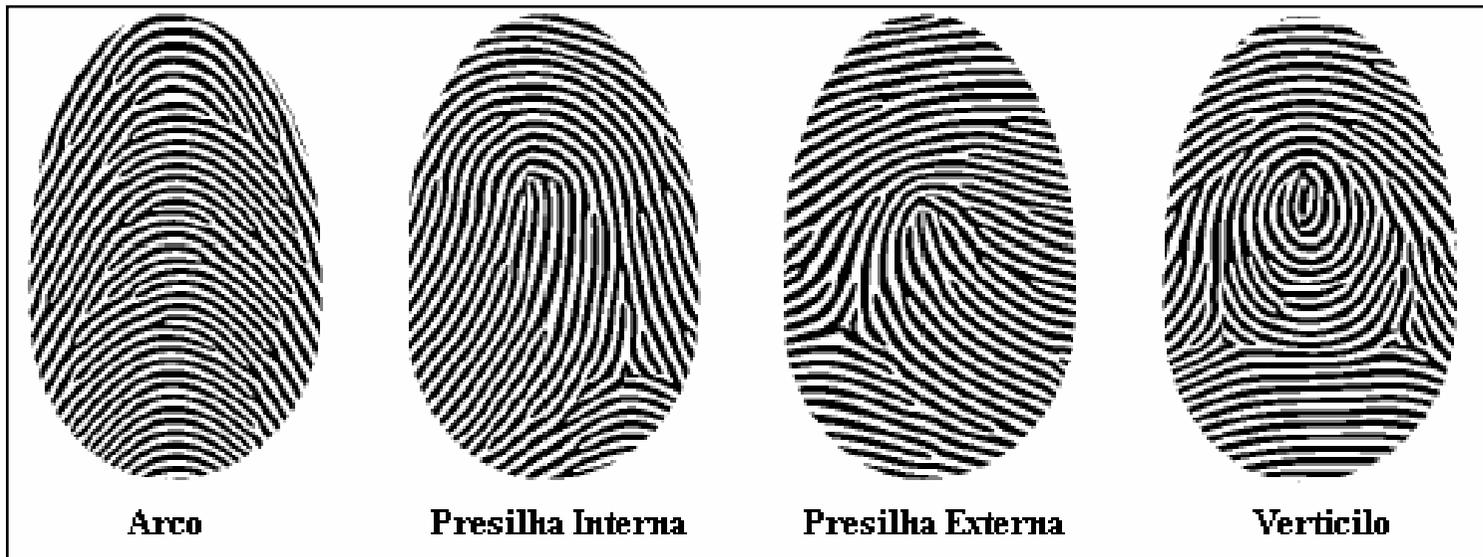
# Impressões Digitais

- ✓ Pontos característicos (linhas pretas, linhas brancas, minúcias, delta e núcleo)



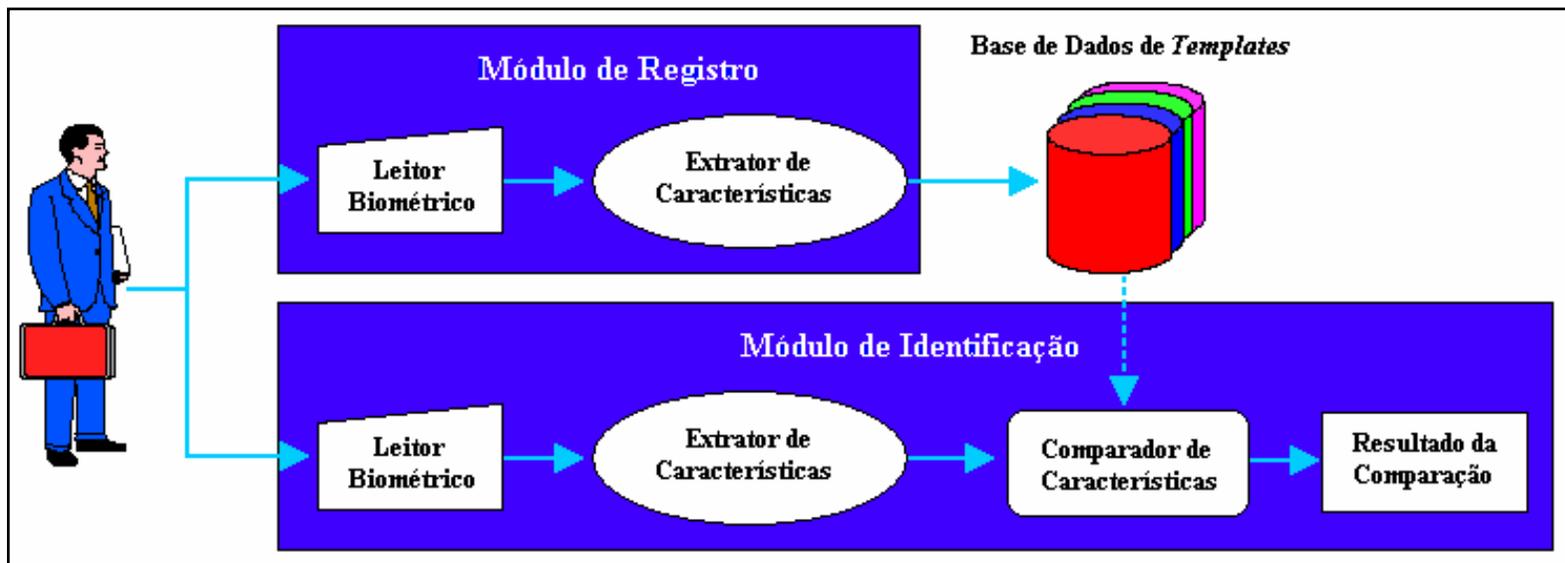
# Impressões Digitais

✓ Classificação no Sistema de Vucetich: tipos fundamentais



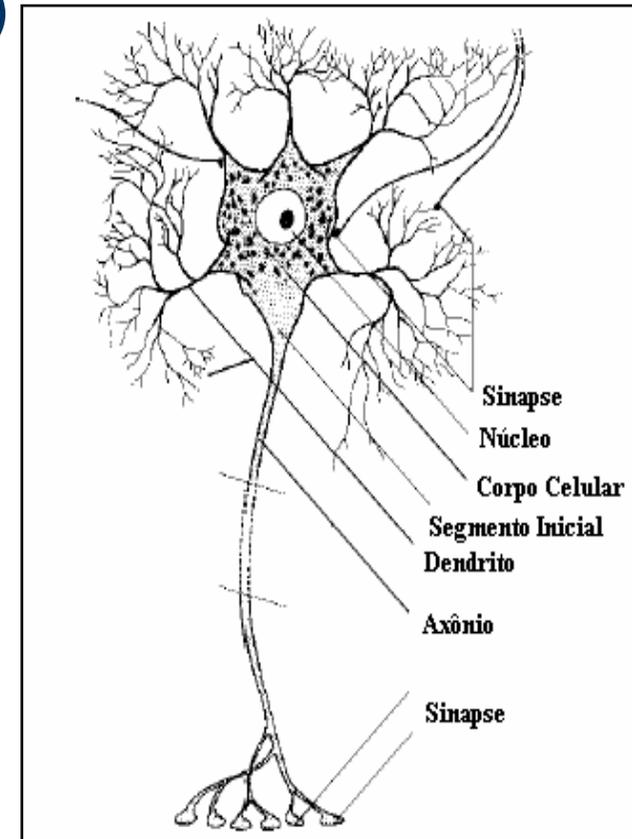
# Biometria

- ✓ Sistemas de identificação biométrica (recente)
- ✓ Reconhecimento através de características físicas, fisiológicas ou comportamentais



# Redes Neurais

- ✓ Cérebro Humano (Capacidades)
- ✓ Quantidade de neurônios
- ✓ Neurônios naturais

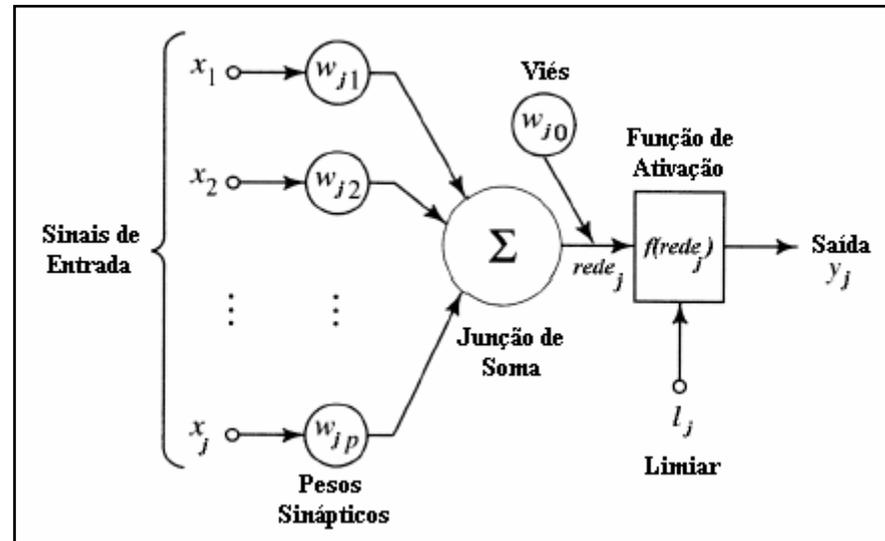


# Redes Neurais Artificiais

- ✓ Início (1940)
- ✓ Conceitos (Simular Rede neural biológica)
- ✓ Modelos (*ADALINE/MADALINE*, *Time-Delay*, *BAM*, etc.)
- ✓ Treinamento (Supervisionado e Não-Supervisionado)

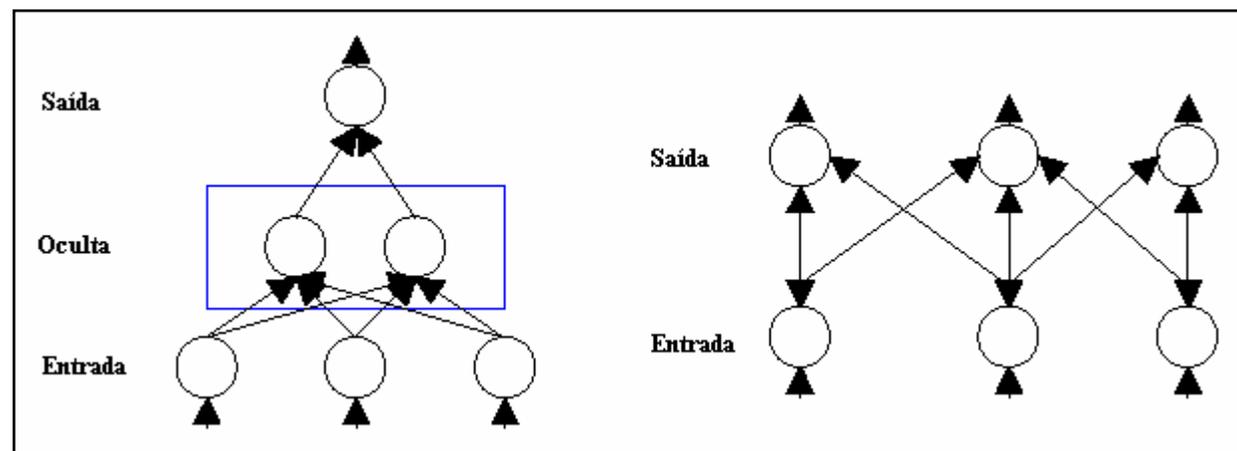
# Neurônio Artificial

- ✓ Comparação com neurônio natural
- ✓ Pesos dos neurônios
- ✓ Disposição em camadas (entrada, oculta e saída)



# Rede Neural Artificial Utilizada

- ✓ *Perceptron Multicamada - FeedFoward:*
  - Baseados em perceptrons
  - Treinamento (*Backpropagation - duas etapas*)
  - Validação

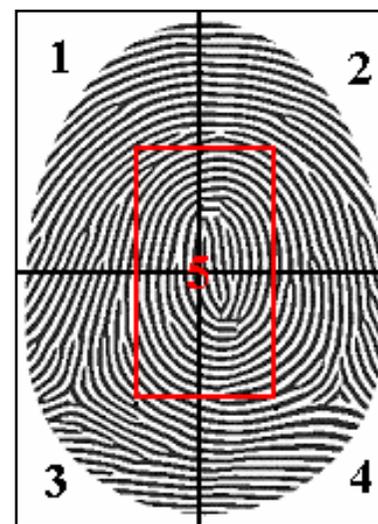
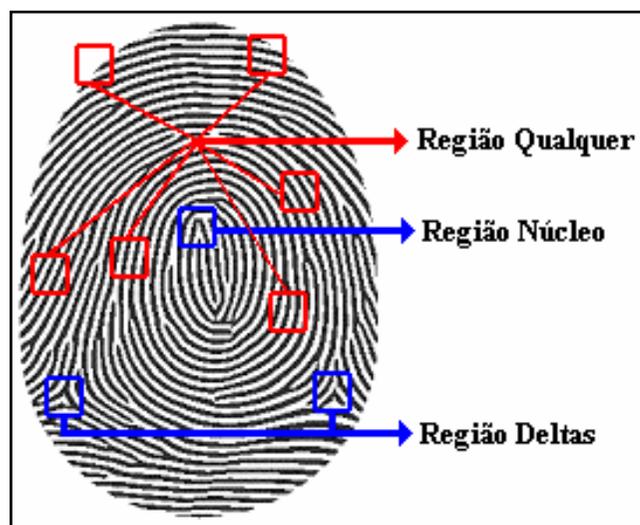


## Trabalhos Correlatos

- ✓ Hong (1998) – *Automatic personal identification using fingerprints*
- ✓ Silva (1999) – Protótipo de Software para Classificação de Impressão Digital
- ✓ Gumz (2002) – Protótipo de um sistema de identificação de minúcias em Impressões Digitais utilizando redes neurais artificiais *feedforward* multicamada

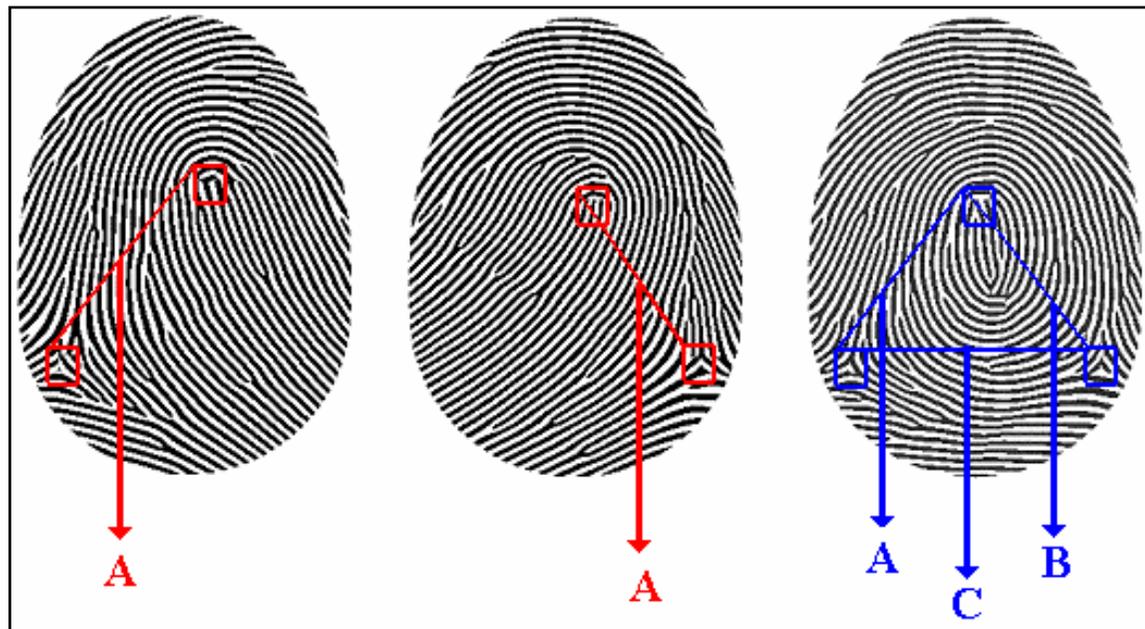
# Desenvolvimento – Visão Geral

- ✓ Definição das regiões para treinamento
- ✓ Definição de quadrantes



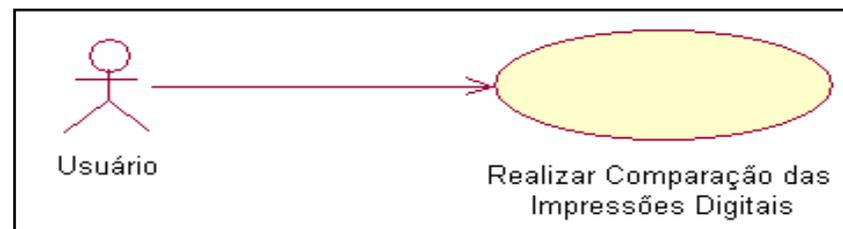
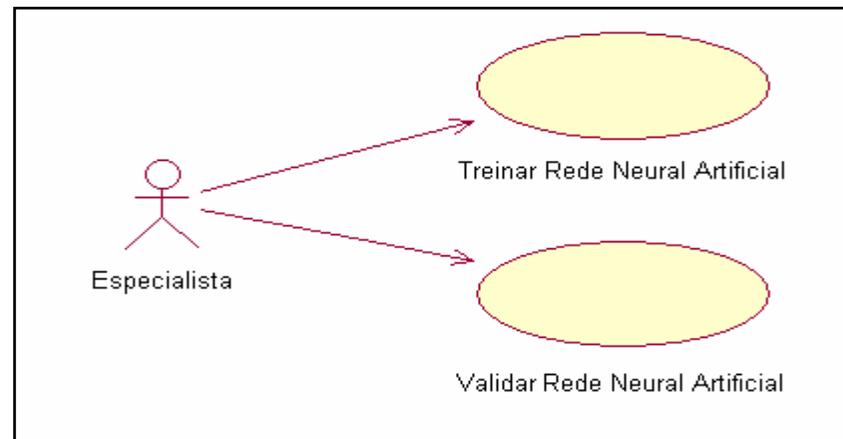
# Desenvolvimento – Visão Geral

- ✓ “Heurística do cálculo da distância”
- ✓ Utilização para 3 dos 4 tipos de impressões digitais



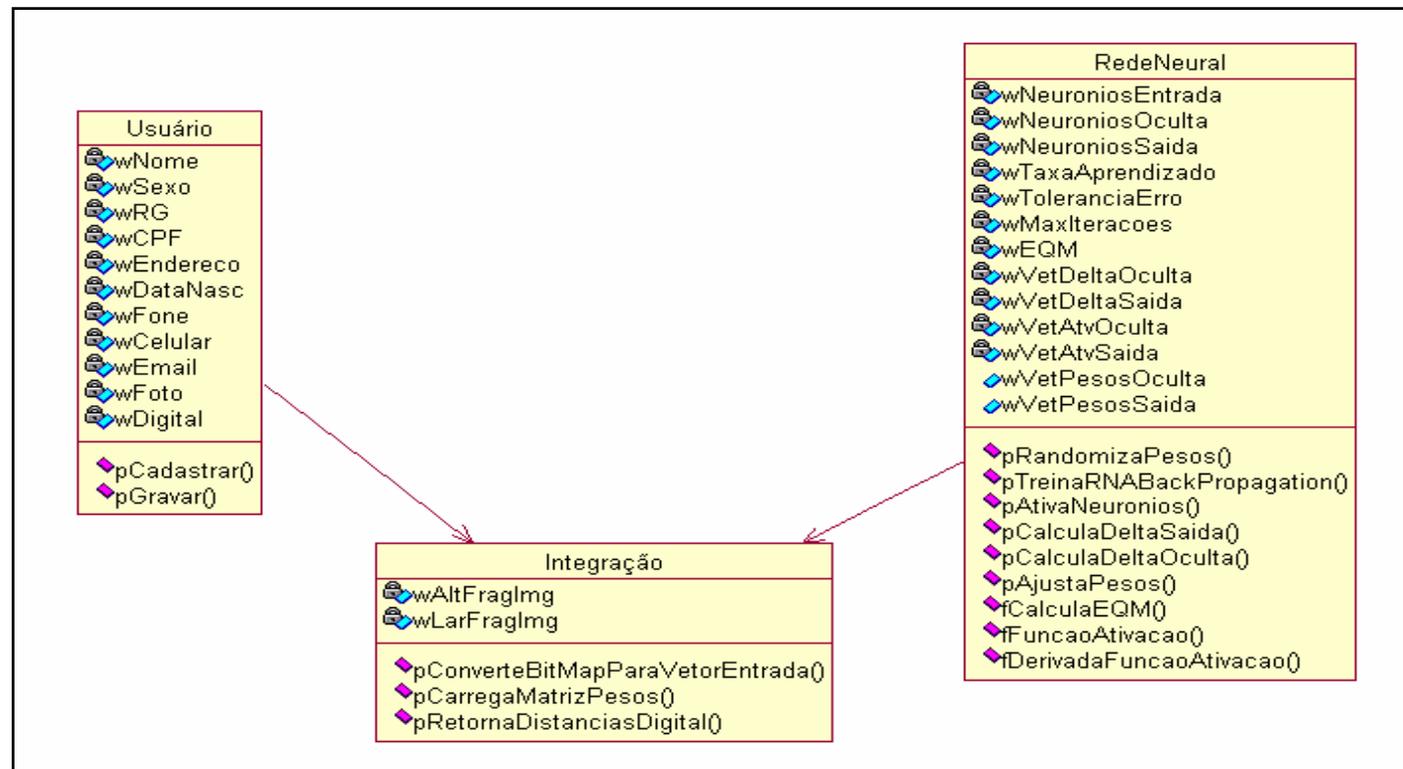
# Especificação – Casos de Uso

- ✓ Casos de uso primários



# Especificação – Diagrama

## ✓ Diagrama de Classes



## Especificação – Parâmetros

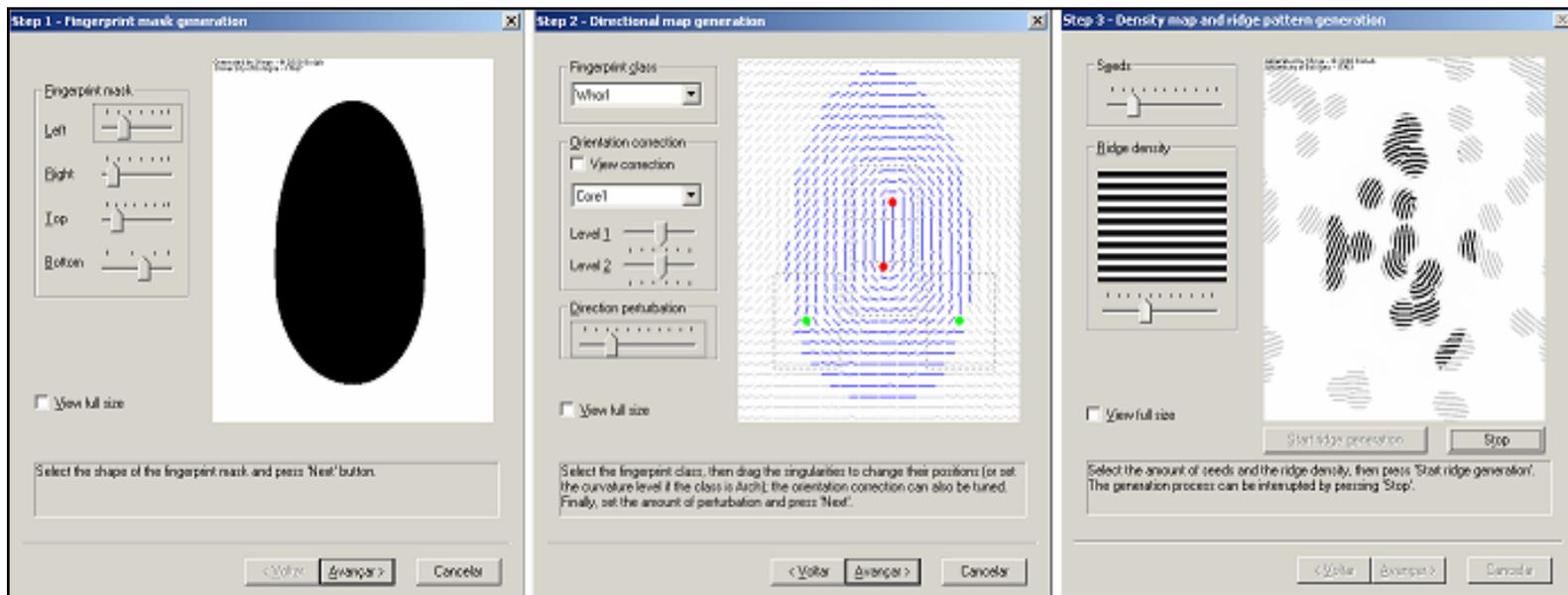
- ✓ Parâmetros de Treinamento da Rede Neural:
  - Camadas (1 camada oculta)
  - Quantidade de neurônios (400, 40, 1)
  - Erro quadrado médio (EQM – 0,0001)
  - Taxa de Aprendizado (0,1)
  - Valores Alvo (0,15 – deltas / 0,45 – núcleos / 0,8 - qualquer)

# Implementação

- ✓ Aquisição de Imagens:
  - Desenhos digitais com tinta
  - Leitores biométricos
  - Internet (Não sabendo a fonte)
  - Software SFinge

# Implementação – SFinge

- ✓ Configurações para geração de imagens artificiais:



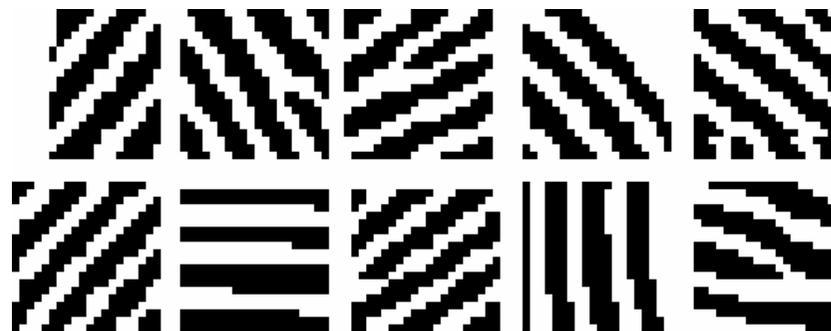
# Implementação – SFinge

- ✓ Figuras geradas pelo Software:



# Implementação – Amostras

- ✓ Deltas e núcleos, regiões quaisquer



# Implementação – *Backpropagation*

```
procedure pTreinarNABackPropagation(prVetEnt, prVetSai: TArray2D; prQtdAmost: Integer;
                                     var prRetPesosOculta: TArray2D; var prRetPesosSaida: TArray2D);
begin
  // Realiza o laço para a quantidade de Iterações
  for wIteracao := 0 to wMaxIteracoes do
    begin
      // Incicializa a Variável de Erro
      wAcumulaEQM := 0;

      // Laço sobre todos as Amostras de Entradas
      for wAmostras := 0 to prQtdAmostras-1 do
        begin
          wVetAtuEntrada := TArray1D(prVetEntradas[wAmostras]);
          wVetAtuSaida := TArray1D(prVetSaidas[wAmostras]);
          pAtivaNeuronios(wNroNodosEnt, wNroNodosOcu, wVetPesosOculta, wVetAtuEntrada, wVetAtivOculta);
          pAtivaNeuronios(wNroNodosOcu, wNeuroniosSai, wVetPesosSaida, wVetAtivOculta, wVetAtivSaida);

          // Chama a função que Calcula o Erro Quadrado Médio
          wEQM := fCalculaEQM;

          // Verifica se precisa ajustar para este caso de treinamento
          if wEQM > wToleranciaErro then
            begin
              // Chama as procedures que calculam os Deltas.
              pCalculaDeltaSaida;
              pCalculaDeltaOculta;
              // Chama as procedures que realizam o ajuste dos pesos nos Vetores de pesos.
              pAjustaPesos(wVetDeltaSai, wVetAtivOcu, wVetPesosSai, wNroNodosSaida, wNroNodosOcu);
              pAjustaPesos(wVetDeltaOcu, wVetAtuEnt, wVetPesosOcu, wNeuroniosOcu, wNeuroniosEnt);
            end;

            wAcumulaEQM := wAcumulaEQM + wEQM;
          end;

          // Realiza a média dos erros
          wAcumulaEQM := wAcumulaEQM / prQtdAmostras;

          // Verificar Tolerância ao Erro
          if (wAcumulaEQM < wToleranciaErro) then
            `*** Rede Treinada com Sucesso ***`;
          end;
        end;
      end;
    end;
end;
```

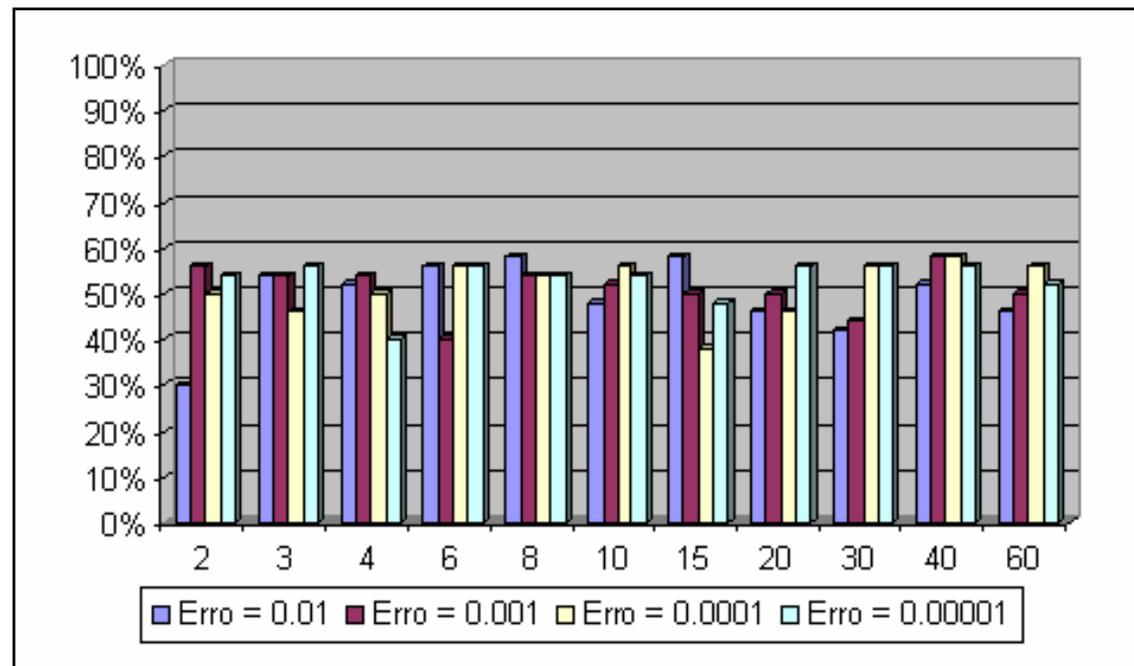
# Resultados

- ✓ Percentual de acerto x Quantidade de NOs (oculta)
- ✓ EQM = 0,0001 e NOs = 40

NO	Erro = 0.01	Erro = 0.001	Erro = 0.0001	Erro = 0.0001
2	30%	56%	50%	54%
3	54%	54%	46%	56%
4	52%	54%	50%	40%
6	56%	40%	56%	56%
8	58%	54%	54%	54%
10	48%	52%	56%	54%
15	58%	50%	38%	48%
20	46%	50%	46%	56%
30	42%	44%	56%	56%
40	52%	58%	58%	56%
60	46%	50%	56%	52%

# Resultados

- ✓ Percentual de acerto x Quantidade de NOs (oculta)
- ✓ EQM = 0.0001 e NOs = 40



# Resultados

## ✓ Grupos de amostras

Tipo de Região	Grupo 1	Grupo 2	Grupo 3	Grupo 4	Grupo 5	Grupo 6	Grupo 7	Grupo 8
Deltas	25	50	100	200	400	800	1.600	3.200
Núcleos	25	50	100	200	400	800	1.600	3.200
Regiões Qualquer	50	100	200	400	800	1.600	3.200	6.400

# Resultados

✓ Ideal > 90%, Obtido = 58%

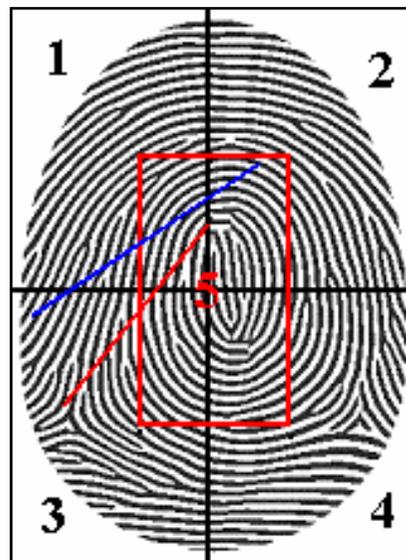
Grupos	Erro = 0.0001
Grupo 1	52%
Grupo 2	52%
Grupo 3	58%

# Resultados

- ✓ Ferramentas
  - Borland Delphi 7.0
  - Rational Rose
  - MS-Paint
  - SFinge
  
- ✓ Redes Neurais Artificiais (desempenho insatisfatório):
  - Pré-processamento
  - Poucas amostras treino
  - Apenas uma Rede Neural

# Conclusão

Criação de um protótipo que atende parcialmente os objetivos iniciais, uma vez que o desempenho da rede se mostrou abaixo do esperado



## Extensões

- ✓ Utilizar leitores biométricos
- ✓ Técnicas de pré-processamento
- ✓ Duas Redes Neurais