

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

**SISTEMA AUTOMATIZADO DE IDENTIFICAÇÃO DE  
IMPRESSÕES DIGITAIS PEER-TO-PEER (P2P)**

**CLAYTON GUIMARÃES COVA DOS SANTOS**

**ORIENTADOR: RAFAEL TIMÓTEO DE SOUSA JÚNIOR**

**DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA  
ÁREA DE CONCENTRAÇÃO INFORMÁTICA FORENSE E  
SEGURANÇA DA INFORMAÇÃO**

**PUBLICAÇÃO: PPGENE.DM – 627 P/16**

**BRASÍLIA / DF: DEZEMBRO/2016**

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

**SISTEMA AUTOMATIZADO DE IDENTIFICAÇÃO DE  
IMPRESSÕES DIGITAIS PEER-TO-PEER (P2P)**

**CLAYTON GUIMARÃES COVA DOS SANTOS**

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 PROFISSIONAL EM INFORMÁTICA FORENSE E SEGURANÇA DA INFORMAÇÃO.

APROVADA POR:



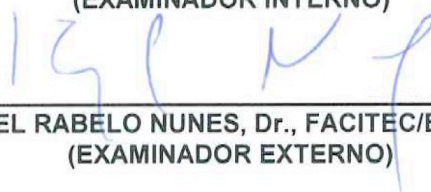
---

**RAFAEL TIMÓTEO DE SOUSA JÚNIOR, Dr., ENE/UNB  
(ORIENTADOR)**



---

**ROBSON DE OLIVEIRA ALBUQUERQUE, Dr., ENE/UNB  
(EXAMINADOR INTERNO)**



---

**RAFAEL RABELO NUNES, Dr., FACITEC/ESTÁCIO  
(EXAMINADOR EXTERNO)**

DATA: BRASÍLIA/DF, 16 DE DEZEMBRO DE 2016.

## FICHA CATALOGRÁFICA

SANTOS, CLAYTON GUIMARÃES COVA DOS SANTOS  
Sistema Automatizado de Identificação de Impressões Digitais Peer-to-Peer (P2P) [Distrito Federal]  
2016.

(XIII), (66) p., 297 mm (ENE/FT/UnB, Mestre, Engenharia Elétrica, 2016).

Dissertação de Mestrado – Universidade de Brasília, Faculdade de Tecnologia. Departamento de Engenharia Elétrica.

1. Identificação, 2. AFIS 3. Sistemas automatizados de identificação, 4. Interoperabilidade, 5. Sistemas peer-to-peer.

I. ENE/FT/UnB. II. Título (Série)

## REFERÊNCIA BIBLIOGRÁFICA

SANTOS, C. G. C. (2016). Sistema Automatizado de Identificação de Impressões Digitais Peer-to-Peer (P2P). Dissertação de Mestrado, Publicação PPGENE.DM – 627 P/16, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, (66) p.

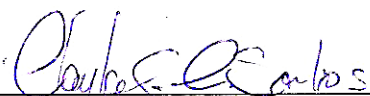
## CESSÃO DE DIREITOS

NOME DO AUTOR: Clayton Guimarães Cova dos Santos

TÍTULO DA DISSERTAÇÃO: Sistema Automatizado de Identificação de Impressões Digitais Peer-to-Peer (P2P).

GRAU/ANO: Mestre/2016.

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. Do mesmo modo, a Universidade de Brasília tem permissão para divulgar este documento em biblioteca virtual, em formato que permita o acesso via redes de comunicação e a reprodução de cópias, desde que protegida a integridade do conteúdo dessas cópias e proibido o acesso a partes isoladas desse conteúdo. O autor reserva outros direitos de publicação e nenhuma parte deste documento pode ser reproduzida sem a autorização por escrito do autor.



Clayton Guimarães Cova dos Santos

Alameda Roquete Pinto, 4482, Residencial ônix, apto.207, bloco A. Bairro: Nova Esperança.

CEP 76822-180 – Porto Velho – RO – Brasil

Dedico este trabalho à minha esposa Emanuely, minha filha Ada, meu pai Manuel e minha mãe Zuila. Também a toda a família, por serem o lastro e arcabouço para a minha vida.

## **AGRADECIMENTOS**

A Deus, por tudo ser possível. Ao meu amigo Dr. Américo Paes da Silva, pelo incentivo, lições e experiências transmitidas, oriundas do seu grande saber. Especialmente ao meu orientador Prof. Dr. Rafael Timóteo de Sousa Júnior, pela inspiração, inteligência, apoio e amizade, elementos essenciais para o desenvolvimento deste trabalho e para o meu desenvolvimento como aluno.

A todos os colaboradores do Programa de Pós-Graduação, representado aqui pelo Prof. Flávio Elias Gomes de Deus, do Curso de Engenharia de Redes de Comunicação - Departamento de Engenharia Elétrica.

Ao Governo do Estado de Rondônia, especialmente à Rosana Cristina Vieira de Souza, Bruno da Silva Pinheiro e Fabio Soares Folly, por terem apoiado e viabilizado muito dos passos necessários para o desenvolvimento deste trabalho.

A todos, os meus sinceros agradecimentos.

O presente trabalho foi realizado com o apoio do Departamento Polícia Federal – DPF ou Instituto de Criminalística do Estado de Rondônia, com recursos do Programa Nacional de Segurança Pública com Cidadania – PRONASCI, do Ministério da Justiça.

## **RESUMO**

### **SISTEMA AUTOMATIZADO DE IDENTIFICAÇÃO DE IMPRESSÕES DIGITAIS PEER-TO-PEER (P2P)**

**Autor:** Clayton Guimarães Cova dos Santos

**Orientador:** Rafael Timóteo de Sousa Júnior

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

**Brasília, setembro de 2016**

Sistemas Automatizados de Identificação de Impressões Digitais (AFIS – *Automated Fingerprint Identification System*) são sistemas que executam confrontos de impressões digitais de forma automatizada, permitindo operações de verificação (1 para 1) ou identificação (1 para  $N$ ) de impressões digitais cadastradas neste ambiente. O trabalho aqui descrito apresenta a concepção de um sistema de localização, compartilhamento e confronto de templates biométricos em sistemas AFIS no formato *peer-to-peer*, utilizando o protocolo *Chord* para distribuição dos identificadores de indivíduos cadastrados. Também apresenta uma proposta de escalonamento para filas de confronto baseada em custos, com métricas e pesos calculados para cada entrada na fila, considerando parâmetros como ordem de chegada, disponibilidade, número de requisições, número de respostas e tamanho da base de *templates*.

Tal solução se apresenta como uma alternativa para confronto entre *templates* de nós distintos, podendo ser utilizado em consultas de verificação ou pesquisas de identificação entre *peers*. Esta aplicação poderia integrar Estados brasileiros, permitindo consultas entre bases biométricas e possibilitando a identificação de indivíduos registrados em outras unidades da federação.

Foi possível constatar através de simulações entre os nós, compondo filas de identificação aleatoriamente, que a fórmula proposta permite um reordenamento dos itens recebidos de maneira satisfatória, escalonando o processamento dos templates no sistema AFIS dos nós participantes. O protocolo *Chord* se mostrou eficiente para viabilizar a localização dos templates, mantendo um índice de pesquisa com base nos CPFs dos prontuários existentes em cada nó da rede.

## **ABSTRACT**

### **PEER-TO-PEER AUTOMATED FINGERPRINT IDENTIFICATION SYSTEM**

**Author: Clayton Guimarães Cova dos Santos**

**Supervisor: Rafael Timóteo de Sousa Júnior**

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

**Brasília, september of 2016**

Automated Fingerprint Identification System executes a comparison operation between fingerprints automatically, allowing thus verification (1 to 1) or identification (1 to  $N$ ) of fingerprints registered at these environments. The work here described shows a concept system to locate, share and match biometric templates in a peer-to-peer network, using the Chord protocol to distribute their identifiers. Present too an scaloning proposal to identification queues based on cost, with metrics and weights defined to every queue entry, resorting parameters like arrive order, disponibility, requisitions number, response number and template database size.

That solution is shown as an alternative to confronting templates from distinct nodes, able to be used in verification operations (1:1) or identification searches (1: $N$ ) between peers. This application could integrate Brazilian states, allowing queries among biometric databases and enabling person identification in other federation units.

It was possible verify through inter nodes simulations, compounding identification queues randomly, that the proposed formula provide a reordering of received items in an efficient way, scaloning templates processing at AFIS of the participant nodes. The Chord protocol proved efficient to make feasible template searches, maintaining a search index based on CPFs information at records of each node.

# SUMÁRIO

|  |    |
|--|----|
| 1. INTRODUÇÃO.....   | 1  |
| 1.1. MOTIVAÇÃO.....  | 1  |
| 1.2. HIPÓTESE .....  | 2  |
| 1.3. CENÁRIO ATUAL.....                                    | 3  |
| 1.4. OBJETIVOS.....  | 4  |
| 1.5. RESULTADOS ESPERADOS.....                             | 5  |
| 1.6. ESTRUTURA DO TRABALHO .....                           | 6  |
| 2. FUNDAMENTOS TEÓRICOS.....                               | 7  |
| 2.1. AFIS .....  | 9  |
| 2.2. INTEROPERABILIDADE ENTRE AFIS.....                    | 14 |
| 2.3. APLICAÇÕES PEER-TO-PEER.....                          | 17 |
| 2.3.1. REPUTAÇÃO EM SISTEMAS PEER-TO-PEER.....             | 18 |
| 2.3.2. MÉTRICAS E CUSTOS EM APLICAÇÕES PEER-TO-PEER.....   | 21 |
| 2.3.3. PROTOCOLO CHORD.....                                | 23 |
| 3. APLICAÇÃO AFIS .....                                    | 28 |
| 3.1.1. CONVERSÃO DE TEMPLATES.....                         | 29 |
| 3.1.2. MÓDULO DE VERIFICAÇÃO.....                          | 31 |
| 3.1.3. MÓDULO DE IDENTIFICAÇÃO.....                        | 33 |
| 4. AFIS PEER-TO-PEER.....                                  | 36 |
| 4.1. PROTOCOLO DE COMUNICAÇÃO.....                         | 36 |
| 4.2. PROCESSAMENTO DE TEMPLATES .....                      | 37 |
| 4.3. CONFRONTO DE VERIFICAÇÃO.....                         | 38 |
| 4.4. CONFRONTO DE IDENTIFICAÇÃO .....                      | 39 |
| 4.5. PRIORIDADE E BALANCEAMENTO DE CARGA NOS NÓS .....     | 40 |
| 5. RESULTADOS .....  | 51 |
| 5.1. LOCALIZAÇÃO DE TEMPLATES .....                        | 54 |
| 5.2. PROPAGAÇÃO DE TEMPLATES E EXECUÇÃO DE CONFRONTOS..... | 56 |
| 5.3. ESCALONAMENTO DA FILA DE IDENTIFICAÇÃO .....          | 58 |



|  |           |
|--|-----------|
| <b>6. CONCLUSÕES.....</b>              | <b>62</b> |
| <b>6.1. TRABALHOS PUBLICADOS .....</b> | <b>63</b> |
| <b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b> | <b>64</b> |

## LISTA DE TABELAS

|   |    |
|---|----|
| Tabela 2.1 – Propriedades de Identificadores Biométricos .....                  | 7  |
| Tabela 4.1 - Fila De Identificação Simulada, Ordenadas Pelo Valor Do Custo..... | 44 |
| Tabela 5.1 - Fila De Identificação Do <i>Host</i> “Ro.Rafis.Net”.....           | 60 |

## LISTA DE FIGURAS

|   |    |
|---|----|
| Imagem 1.1 - Estrutura lógica de comunicação da rede AFIS <i>peer-to-peer</i> .....   | 5  |
| Imagem 2.1- Pontos utilizados pelo INI para confronto das impressões digitais [Vieira, 2015].<br>.....  | 8  |
| Imagem 2.2 - Pares de minúcias confrontadas, exibindo as propriedades de distância entre as minúcias com as retas projetadas e os ângulos formados entre os pontos marcados [Watson <i>et al</i> , 2007]. .....   | 11 |
| Imagem 2.3 - Minúcias marcadas na impressão digital com o polígono respectivo formado, minúcias 1 e 2 destacadas com os respectivos subgrafos apresentados.....   | 12 |
| Imagem 2.4 - Apresenta as principais técnicas de extração de características. [Maltoni <i>et al</i> , 2005] .....   | 13 |
| Imagem 2.5 - Apresenta conjunto das técnicas de classificação de impressões digitais [Dorizzi <i>et al</i> , 2009]. .....   | 13 |
| Imagem 2.6 - Anel lógico de identificação com $m$ igual a 3 bits, estando os nós 0, 1 e 3 inseridos na rede. Neste exemplo a chave 1 é posicionada no nó 1, a chave 2 no nó 3 e a chave 6 no nó 0, sempre buscando o nó sucessor à chave inserida [Stoica <i>et al</i> , 2001]. ..... | 24 |
| Imagem 2.7 - <i>Finger table</i> do nó N8 apresentando os nós sucessores para chaves do anel. ....  | 24 |
| Imagem 3.1 - Janela de status de conversão e inserção dos <i>templates</i> no banco.....  | 30 |
| Imagem 3.2 - Exemplar de <i>template</i> XML criados.....   | 30 |
| Imagem 3.3 - Conjunto de classes da API do <i>SourceAFIS</i> . .....  | 31 |
| Imagem 3.4 - Janela de variáveis locais do Visual Studio detalhando o objeto “pessoa01” com seus respectivos atributos.....   | 32 |
| Imagem 3.5 - Tela do módulo de identificação, tendo a imagem de impressão revelada submetida à pesquisa 1:N, retornando o candidato com maior pontuação, positivando a impressão questionada. ....  | 34 |
| Imagem 4.1 - Sequência lógica de comunicação entre os nós remoto e proprietário na operação de verificação da rede AFIS <i>peer-to-peer</i> .....   | 39 |
| Imagem 4.2 - Regressão do valor do custo frente ao número de respostas recebidas de cada nó.<br>.....   | 46 |
| Imagem 4.3 - Projeção comparativa entre o custo padrão e o custo calculado sem considerar o Número de Respostas – NR. ....  | 46 |
| Imagem 4.4 - Regressão do valor do custo frente aos valores de disponibilidade (DN) de cada nó.....   | 47 |

|   |    |
|---|----|
| Imagem 4.5 - Valores de custo para cada entrada da tabela frente aos valores de Disponibilidade do Nó – DN.....   | 47 |
| Imagem 4.6 - Gráfico ilustra o valor do custo frente a variação na Ordem de Chegada de cada entrada. Percebe-se pela distribuição que grande parte das entradas têm sua posição redefinida se comparada a ordem de chegada inicial.....               | 48 |
| Imagem 4.7 - Plotagem do custo frente ao custo recalculado, sem considerar na fórmula o valor da Ordem de Chagada – OC. ....  | 48 |
| Imagem 4.8 - Representação do custo frente aos dados de Número de Requisições (NQ) de cada entrada. Também a variação do cálculo de custo normal e o custo sem considerar a variável NQ. ....   | 49 |
| Imagem 4.9 - Custos calculados para a fórmula proposta frente ao recálculo do custo sem considerar o tamanho da Base de Dados. O gráfico de barras contrasta com os demais pontos, apresentando o tamanho da base de cada nó da entrada na fila. .... | 49 |
| Imagem 5.1 - Esquema simplificado dos módulos e tabelas criadas para o funcionamento do AFIS <i>peer-to-peer</i> . ....   | 51 |
| Imagem 5.2 - Interface desenvolvida para controlar o processo “Rafis” e tela de monitoramento das consultas, fila de processamento e cliente para consulta de <i>templates</i> . ....   | 54 |
| Imagem 5.3 - Versão do módulo “ <i>Rchord</i> ” em linha de comando, utilizado para inserção de templates e nós virtuais, apresentando a pesquisa e localização de prontuários pelo mapeamento CPF, FQDN. ....  | 55 |
| Imagem 5.4 - Seção correspondente ao cliente da aplicação AFIS <i>peer-to-peer</i> , na tela principal da ferramenta. ....  | 56 |
| Imagem 5.5 - Conteúdo constatado nos arquivos de <i>log</i> , apresentando o envio e o recebimento do <i>template</i> submetido a consulta. ....  | 57 |
| Imagem 5.6 - Conteúdo constatado no arquivo “RafisCore.log”, apresentando o resultado do confronto de Verificação.....  | 58 |
| Imagem 5.7 - Tela de monitoramento da tabela de consultas do nó “AM”, constando o resultado de solicitações para o nó “RO”. ....  | 58 |
| Imagem 5.8 - Esquema apresentando a rede <i>peer-to-peer</i> , com os nós participantes representando Estados.....  | 59 |
| Imagem 5.9 - Tela de consulta no <i>host</i> “AM.rafis.net” configurada para enviar consulta ao <i>host</i> “RO.rafis.net”. ....  | 59 |
| Imagem 5.10 - Tabela de métricas dos nós com os respectivos valores das variáveis durante a operação. ....  | 61 |

## LISTA DE SÍMBOLOS, NOMENCLATURA E ABREVIACÕES

AFIS - *Automated Fingerprint Identification System*

ANSI - *American National Standards Institute*

API - *Application Programming Interface*

CJIS - *Criminal Justice Information Services*

CPF - *Cadastro de Pessoas Físicas*

DHT – *Distributed Hash Tables*

EBTS – *Eletronic Biometric Transmission Specification*

EER - *Equal Error Rate*

EFS - *Extended Feature Set*

FBI - *Federal Bureau of Investigation*

FQDN - *Fully Qualified Domain Name*

IAI – *International Association for Identification*

IBGE - *Instituto Brasileiro de Geografia e Estatística*

IEC - *International Electrotechnical Commission*

INCITS - *The InterNational Committee for Information Technology Standards*

INI - *Instituto Nacional de Identificação*

IP - *Internet Protocol*

ISO - *International Organization for Standardization*

ITL - *Information Technology Laboratory*

JPEG - *Joint Photographics Experts Group*

JSON - *JavaScript Object Notation*

NAS – *National Academy of Science*

NIJ – *National Institute of Justice*

NIST – *National Institute of Standards and Technology*

RCN - *Registro Civil Nacional*

SESP - *Secretaria de Estado da Segurança Pública*

SGBD - *Sistema de Gerenciamento de Banco de Dados*

TCP - *Transmission Control Protocol*

WSQ - *Wavelet Scalar Quantization*

XML - *Extensible Markup Language*



## **1. INTRODUÇÃO**

A crescente incidência de crimes na sociedade causa nas instituições de segurança do país um elevado volume de trabalho, em todas as instâncias de Governo. Por isso, há uma necessidade imperativa de que a resposta para a sociedade seja ágil e que as instituições de investigação se tornem mais eficientes, frente a esta corrente demanda. Desta forma, entende-se que, procedimentos periciais, que neste íterim executam a materialidade do fato delituoso, devem estar estruturadas e tecnicamente preparadas para o tratamento desta volumosa carga de trabalho.

A atuação do Perito em campo, durante exames em locais de crime, culmina, dentre outras atividades, na coleta de variada gama de material para análise minuciosa e detalhada, de caráter interno em seções e laboratórios destas instituições técnico-científicas. Dentre os materiais arrecadados como vestígios estão impressões papilares ou fragmentos destas, agregados à objetos da cena do crime ou apostos em suportes diversos. Estas informações biométricas apostas no local ligam o infrator ao sítio do crime, indicando sua possível participação na dinâmica delituosa.

Hoje no Brasil, cada Estado define de forma autônoma como exercerá a gestão e a emissão dos registros de identificação de seus cidadãos. Os sistemas utilizados para viabilizar este trabalho são contratados de forma independente e a administração destas informações representa poder político e estratégico para os Governos Estaduais. A dificuldade de se criar uma identidade nacional, como exemplo, conflita via de regra com os interesses e resistência dos órgãos nestes Estados e seus políticos, principalmente quando o tema da proposta vislumbra o compartilhamento ou transferência de informações importantes, como dados de cidadãos.

### **1.1. MOTIVAÇÃO**

Atualmente não existe uma solução ou sistema automatizado de confronto de impressões digitais que viabilize a consulta centralizada de informações biométricas em um nível nacional e que permita aos órgãos de segurança pública a execução de verificações de identidades emitidas entre estados brasileiros. Tal desajustamento inviabiliza a utilização inteligente destas informações, impedindo a pesquisa de prontuários biométricos para os mais diversos fins. Esse desarranjo nos processos de comunicação e compartilhamento de informações estatais prejudicam especialmente as atividades dos órgãos de segurança pública.

Desta forma, frente ao montante de informações que são adquiridas em locais examinados em perícias feitas, o Brasil ainda não dispõe de uma infraestrutura própria padronizada para a gestão e processamento destes elementos coletados, em especial dos vestígios papiloscópicos. Sabendo que a atuação criminosa de indivíduos não se restringe a limites geopolíticos, torna-se interessante aos organismos de segurança possuírem dados que permitam a identificação de pessoas oriundas dos demais regiões federadas.

Considerando as prerrogativas e a autonomia das unidades membros da Federação Brasileira, a ausência de um protocolo de comunicação entre bases biométricas no contexto nacional coloca em evidência a necessidade de uma alternativa tecnológica para a troca de informações e execução de pesquisa entre as bases estaduais de identificação criminal. Contudo, sem uma via padronizada de comunicação a ser seguida, cada órgão de segurança adota hoje um procedimento distinto para recebimento de solicitações de informações e consultas de outros Estados, além dos procedimentos administrativos, manuais e burocráticos em sua maioria.

As ferramentas e sistemas computacionais hoje disponíveis permitem o tratamento e a organização de todo tipo de informação no espectro do conhecimento humano, resultado da evolução das soluções tecnológicas e da integração dos sistemas e aplicações automatizadas com os dados do mundo real. Tais instrumentos são atualmente utilizados de forma canhestra pela polícia na gestão das informações coletadas em locais de crime, componentes do corpo do delito. Esta deficiência é limitadora para que a visão macro da criminalística esteja ao alcance da gestão e dos órgãos de segurança do Estado.

## **1.2. HIPÓTESE**

Assim sendo, uma vez definido uma forma ou padrão de comunicação entre os órgãos de segurança dos Estados, a troca de informações biométricas de forma automatizada permitiria agilidade e eficiência no processamento de requisições de confrontos biométricos interestaduais, aumentando a capacidade de investigação dos seus órgãos de segurança e resolução de crimes, maximizando o uso das informações de identificação, agregando assim, eficiência na persecução criminal e à Justiça.



### **1.3. CENÁRIO ATUAL**

A Lei nº 9.454 de 07 de abril de 1997 estabeleceu o Registro de Identidade Civil, esse novo documento tinha como objetivo principal a institucionalização de um novo arcabouço de identificação, que integra todos os Estados brasileiros e o Distrito Federal. O Decreto nº 7.166 de 05 de maio de 2010 regulamentou esta lei e estabeleceu o Sistema Nacional de Registro de Identificação Civil – SINRIC e seu respectivo Comitê Gestor. Esta iniciativa tinha como missão a garantia de identificação civil nacional confiável a todo brasileiro, por meio de biometria e biografia, visando ser instrumento de cidadania tornando mais seguras e eficazes suas relações com o Estado e a sociedade [BRASIL, 2016]. Este projeto teve seus estudos técnicos e pesquisa suspensos em julho de 2015, devido a proposta apresentada pelo Poder Executivo e pelo Tribunal Superior Eleitoral através do Projeto de Lei nº 1.775, de 28 de maio de 2015, o qual especifica o Registro Civil Nacional – RCN, proposta a ser implementada pelo Tribunal Superior Eleitoral – TSE.

Como um programa de cadastramento mais abrangente, o RCN (Registro Civil Nacional) da Justiça Eleitoral, objetiva em seu escopo o cadastramento biométrico e a verificação da identidade de cidadãos brasileiros, permitindo sua identificação nas relações sociais com o governo e entidades privadas, englobando emissão de documentação com numeração única nacional [TSE, 2014]. Os benefícios apontados alcançam as áreas, do social, da segurança, tributária, previdência social e educação. Já o registro único permitiria a identificação inequívoca do cidadão nos mais diversos sistemas utilizados pelo Estado.

Para a utilização desta infraestrutura de gerenciamento e o acesso aos dados de identificação nacional por parte dos Estados, necessitará de um acordo de cooperação e legislação específica para o compartilhamento de tais informações. Os Governos dos Estados ficarão dependentes do serviço prestado pelo TRE para utilização das informações biométricas em suas soluções personalizadas. Parte das aplicações e soluções tecnológicas já adquiridas e em operação continuariam a ser utilizadas em paralelo, promovendo serviços importantes ao Estado, ainda que o sistema RCN estivesse disponível para utilização. Desta forma, a continuidade na gestão de dados de identificação tende a continuar sendo desenvolvida e aperfeiçoada pelos Estados, principalmente no âmbito da segurança pública, que em sua característica investigativa e de gestão deste tipo de dados, promovendo maior controle sobre as suas populações carcerárias e indiciados por crimes diversos.

Neste ínterim, observa-se que as unidades federadas continuariam alimentando suas bases de dados com informações biométricas, gerenciando e aperfeiçoando suas soluções a fim de atender as mais diversas demandas de interesse de seus gestores. No campo da segurança pública, um controle mais eficiente e universal da população carcerária, por exemplo, significará melhor controle populacional e um fortalecimento das bases criminais, importantes para elucidação de crimes frente ao significativo volume de reincidência de crimes por parte dos infratores [CNJ, 2015].

#### **1.4. OBJETIVOS**

Avaliadas tais condicionantes, o presente trabalho tem por objetivo geral apresentar uma solução distribuída para compartilhamento de informações e interoperação entre distintas bases de dados e sistemas AFIS (*Automated Fingerprint Identification System*), apresentando o sistema e a validação de suas funcionalidades.

Para alcançar o objetivo geral definido, foram especificados os seguintes objetivos específicos:

- Implementar uma ferramenta AFIS, baseada em algoritmos e projeto de código livre. Tal solução executará o confronto de *templates* biométricos no formato ISO;
- Desenvolver uma solução de localização de prontuários biométricos em uma arquitetura de rede *peer-to-peer*, utilizando o protocolo *Chord* como instrumento para a propagação dos identificadores, baseados em CPF;
- Projetar e implementar mecanismos para transferência de *templates* biométricos entre os nós da rede, viabilizando a consulta e o confronto de prontuários em nós distintos;
- Elaborar e implementar uma função de escalonamento para filas de identificação (1:N) dos nós, considerando métricas que viabilizem um modelo justo e sustentável para o processamento de *templates* em todos os nós.
- Examinar o funcionamento e os resultados de confrontos biométricos a partir da prova de conceito concebida, simulando operações de pesquisa, consulta e transferência de *templates* em ambientes AFIS de nós distintos.

## 1.5. RESULTADOS ESPERADOS

Para interoperar sistemas AFIS compartilhando, por exemplo, bases de dados criminais, a proposta aqui apresentada objetiva integrar um sistema de localização de *templates* biométricos em uma rede P2P utilizando o protocolo Chord do MIT [Stoica *et al*, 2001], permitindo a distribuição de um índice de pesquisa em uma estrutura lógica descentralizada. Também propõe uma maneira de escalonar as filas de confrontos biométricos entre os nós, atribuindo custos baseados em informações dos *peers* que solicitam os confrontos.

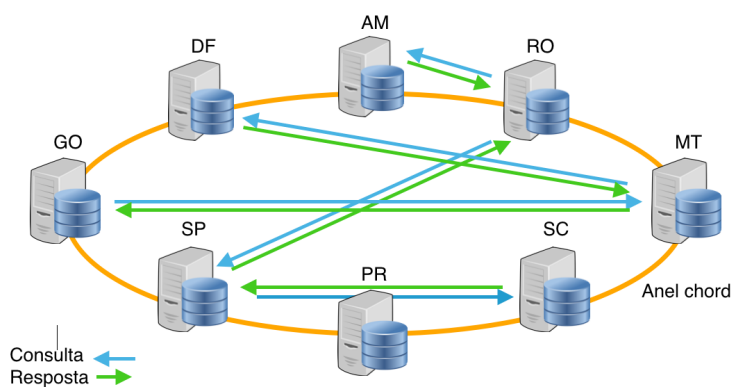


Imagem 1.1 - Estrutura lógica de comunicação da rede AFIS *peer-to-peer*.

Para um Estado operar o sistema proposto como nó, em um primeiro momento, implantado o cliente do AFIS, ele deverá construir uma sua base de dados de identificadores, alimentando esta tabela com todos os registros existentes na sua base de dados criminais. Cada indivíduo existente no banco AFIS será mapeado para o ambiente em rede, recebendo uma chave definida como ID único na rede *peer-to-peer*, na qual o identificador baseado no valor numérico do CPF (Cadastro de Pessoas Físicas) estará associado ao prontuário do indivíduo no banco ao Estado a que ele pertence. A partir de então, as funcionalidades da rede *peer-to-peer* são concebidas para permitir a interoperação entre os AFIS participantes (Figura 1).

Os resultados esperados deste trabalho orbitam três hipóteses principais, que representam o funcionamento probado da solução desenvolvida. A primeira está relacionada com a correta propagação dos identificadores de prontuários nos *peers* da rede, por meio do protocolo Chord. O segundo ponto demanda a eficiência na transmissão e confronto dos prontuários entre os nós, baseando-se nas informações de localização disponibilizadas pelos pontos da rede. Já a terceira hipótese está focada na análise da fila de confronto de identificação (operação 1 para n) no ambiente desenvolvido, considerando as métricas e a fórmula proposta para definição da prioridade de processamento.

## 1.6. ESTRUTURA DO TRABALHO

A sequência do trabalho será apresentada da seguinte forma, no capítulo 2 abordamos os conceitos teóricos do trabalho, explanando acerca da tecnologia dos sistemas AFIS, explicando o conceito, histórico e aplicações disponíveis atualmente. Serão tratados também neste capítulo o protocolo Chord e o seu funcionamento como sistema de mapeamento chave valor para suporte da pesquisa em redes *peer-to-peer*, e o funcionamento do algoritmo AFIS utilizado com base no projeto SourceAFIS. São levantados também técnicas e métodos para definição de métricas de classificação e balanceamento de carga em nós de redes *peer-to-peer*, considerando os protocolos mais utilizados e formatos mais eficientes, apontados pela comunidade científica.

No capítulo 3 apresentamos a proposta base desta dissertação, discorrendo sobre o formato e procedimento de comunicação entre os nós, processamento de *templates*, os confrontos de verificação ( $1:1$ ) e identificação ( $1:n$ ) neste ambiente, bem como a aplicação das métricas de prioridade e balanceamento de carga no processamento de identificação, influenciando a organização e o escalonamento da fila de confronto 1 para  $n$ .

No capítulo 4 são apresentados os resultados das operações, com foco na execução da aplicação, afetos às três hipóteses apresentadas no tópico anterior, verificando a resultante das operações e confrontando as saídas dos testes com os objetivos inicialmente levantados.

O capítulo 5 conclui o trabalho com a diagnose da observação dos testes efetuados, apontando as aplicações a serem contempladas com a solução testada, além das possibilidades de melhorias e propostas para trabalhos futuros, derivados do estudo aqui desenvolvido.

## 2. FUNDAMENTOS TEÓRICOS

Dentre as formas de identificação de indivíduos por suas características biométricas, as impressões papilares, especificamente digitais, se diferem por sua facilidade de coleta, baixo custo e boa confiabilidade no processo de individualização de uma pessoa. A Tabela 1 apresenta uma classificação dos indicadores biométricos, conforme o nível de atendimento (alto, médio, baixo) de cada indicador a um determinado conjunto requisitos de aplicação. Características como universalidade, unicidade, permanência e facilidade de coleta são fatores que colocam em relevo as impressões digitais, apresentando-se como a forma mais acessível e amplamente utilizada para a identificação de indivíduos de forma automatizada [Adderley *et al* 2008].

**Tabela 2.1 – Propriedades de Identificadores Biométricos**

| <b>Identificador Biométrico</b> | <b>1</b> | <b>2</b> | <b>3</b> | <b>4</b> | <b>5</b> | <b>6</b> | <b>7</b> |
|---------------------------------|----------|----------|----------|----------|----------|----------|----------|
| DNA                             | H        | H        | H        | L        | H        | L        | L        |
| Iris                            | H        | H        | H        | M        | H        | L        | L        |
| Retina                          | H        | H        | M        | L        | H        | L        | L        |
| Orelha                          | M        | M        | H        | M        | M        | H        | M        |
| Face                            | H        | L        | M        | H        | L        | H        | H        |
| Impressão Digital               | M        | H        | H        | M        | H        | M        | M        |
| Geometria da Mão                | M        | M        | M        | H        | M        | M        | M        |
| Escrita                         | L        | L        | L        | H        | L        | H        | H        |
| Voz                             | M        | L        | L        | M        | L        | H        | H        |

1: Universalidade; 2: Unicidade; 3: Permanência; 4: Colectabilidade; 5: Desempenho; 6: Aceitabilidade; 7: Suscetibilidade; L: Low (Baixo); M: Medium (Médio); H: High (Alto) [Adderley *et al* 2008] [Tada 2011]

Dentre as técnicas de identificação biométrica listadas na Tabela 1, a impressão digital permite uma maior versatilidade de aplicações. Suas características são extremamente relevantes para os trabalhos afetos à Segurança Pública e à Justiça, visto que na ocorrência de

delitos a interação do infrator com o ambiente muitas vezes deixa vestígios papilares, derivados da dinâmica do ilícito no local em que o crime ocorreu.

Entretanto, o processo de confronto de impressões papilares, se feita de maneira manual, torna-se impraticável nos dias de hoje, cabendo observar em especial que a busca das digitais de suspeitos nos bancos de identificação do Estado se torna inviável e dispendiosa se considerarmos os recursos tecnológicos disponíveis para este trabalho atualmente.

A constatação de impressões papilares em locais de crime é de importância distinta, já que elas representam prova irrefutável contra o infrator [Ludwig 1996]. Tal constatação coloca o agente do delito no local do fato ou mesmo garante a não relação de um suspeito com o crime. Nas impressões digitais as características e propriedades consideradas para classificação são geralmente organizadas em três níveis:

- **Nível 1 (Global):** relacionado ao fluxo das linhas das cristas papilares, orientação, e as propriedades derivadas deste.
- **Nível 2 (Local):** considera as minúcias extraídas do formato do esqueleto das linhas;
- **Nível 3 (Detalhe fino):** incluem detalhes da zona entre as cristas papilares, como largura, contorno, poros e rugas.

As características relativas ao nível 1 são utilizadas geralmente para a classificação das impressões (delta e núcleo), enquanto as de nível 2 e 3 são utilizadas para o confronto com outras impressões [Moses *et al*, 2010]. O padrão que é estabelecido pelo Instituto Nacional de Identificação (INI) da Polícia Federal para o processo de comparação e confronto entre impressões, considera os seguintes pontos característicos: Ponto, Ilhota, Ponta de Linha, Bifurcação, Encerro;

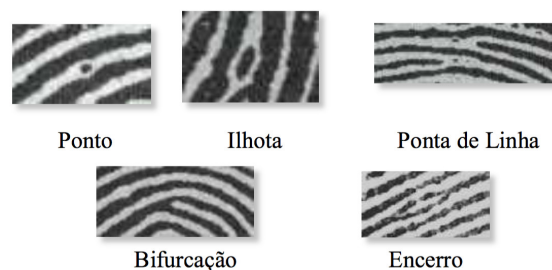


Imagem 2.1- Pontos utilizados pelo INI para confronto das impressões digitais [Vieira, 2015].

As necessidades de automação, destas operações de identificação, justificaram o investimento no desenvolvimento de tecnologia de suporte ao processo de extração e confronto de impressões digitais e suas características, as denominadas minúcias. Minúcia, segundo

Rodrigues *et al* (2011), é um evento anormal presente no fluxo das linhas dactilares que por padrão correm paralelas umas às outras. Estes elementos individualizadores permitem que impressões sejam confrontadas e seu grau de similaridade definido.

Sistemas automatizados para esta tarefa foram desenvolvidos e atualmente são aperfeiçoados com intuito de apresentar resultados mais precisos e rápidos frente ao progressivo volume de informações do mundo moderno.

Muitas das soluções disponíveis hoje para automação deste trabalho de identificação são desenvolvidos na iniciativa privada, originando produtos e sendo negociados como ferramentas de software a elevados custos. Os valores significativos deste tipo de tecnologia impossibilitam muitos Estados da Federação a aquisição das aplicações para o auxílio das forças de segurança pública locais. Quando ocorre a aquisição, tal solução atende de forma localizada, parcial e limitada, consideradas as necessidades reais de trabalho das instituições. Respondendo somente uma pequena parcela da demanda, restritos por número de licenças e pela dependência tecnológica.

## **2.1. AFIS**

Na década de 1960, o FBI nos Estados Unidos, o Home Office do Reino Unido, a Polícia de Paris na França e a Polícia Nacional Japonesa iniciaram projetos para o desenvolvimento de sistemas de automação para identificação de impressões digitais, sistemas atualmente conhecidos como AFIS – *Automated Fingerprint Identification System* [Usmani *et al*, 2013].

Em 1963, o agente especial Carl Voelker em parceria com os engenheiros Raymond Moore e Joe Wegstein do NIST – *National Institute of Standards and Technology*, iniciaram os estudos de viabilização de uma solução computacional para confronto de impressões, através da fomentação do desenvolvimento de equipamentos para escaneamento das papilas, detecção e identificação das minúcias em imagens de digitais, resultando assim no desenvolvimento de um método para comparar dois conjuntos de minúcias e determinar se estes são do mesmo dedo de um indivíduo [Usmani *et al*, 2013].

Um dos primeiros trabalhos relevantes na área de classificação de impressões digitais, utilizando reconhecimento de padrões de forma automatizada foi apresentado no ano de 1975 por B. Moayer e K. S. Fu, através do artigo “*A syntactic approach to fingerprint pattern recognition*”, o qual apresenta uma abordagem sintática para o reconhecimento de padrões em

impressões digitais, subdividindo imagens das impressões em pequenos quadros e processando suas características bloco a bloco [Moses *et al*, 2010].

Atualmente a área de reconhecimento de padrões biométricos, especialmente impressões digitais, possui grande volume de trabalhos publicados e uma infinidade de técnicas e abordagens que permitem o desenvolvimento de diversas soluções para a identificação e confronto de impressões de indivíduos e bases de dados, função que um sistema automatizado executa na sua essência.

Basicamente, o trabalho de um AFIS consiste em três funções principais [Usmani *et al*, 2013]:

- **Registro:** procedimento de inserção das informações de indivíduos na base de dados. Durante o processo de registro, serão coletadas as impressões digitais do indivíduo e o sistema deverá efetuar a verificação da qualidade das impressões capturadas e, após o armazenamento das imagens em um formato padronizado, executar a extração dos pontos característicos compondo o arquivo de minúcias, *template* da impressão. Nesta fase poderão ser inseridas também informações adicionais referentes ao indivíduo cadastrado, de acordo com o projeto proposto em cada modelo de aplicação.
- **Verificação:** processo de verificação de identidade de um indivíduo através do confronto da impressão digital do indivíduo a ser verificado com um registro existente no banco, na forma de 1 para 1. Os dados utilizados para a busca do registro no banco poderão ser informações de cadastro ou documentos como CPF ou RG, que permitirão a consulta do *template* correspondente à impressão que será confrontada.
- **Identificação:** processo para determinar a identificação de um indivíduo por meio de informações biométricas, por exemplo impressões papilares, é efetivado através da extração das informações biométricas, características (minúcias) que formam um *template* para confronto no banco, comparando com todos os outros *templates* ou com um subconjunto do mesmo, na forma 1 para  $n$ . Dependendo da abordagem, o custo de pesquisa dentro do universo de dados do sistema pode aumentar de forma proporcional ao tamanho do conjunto utilizado na pesquisa, por este motivo a classificação dos *templates* em subconjuntos de características/tipo pode melhorar a eficiência da implementação. O resultado da operação poderá retornar uma lista de compatíveis (candidatos) ou uma lista vazia, sem correspondentes [Važan, 2012].



A padronização das tecnologias de codificação e armazenamento dos dados de impressões digitais foi desenvolvida por entidades nacionais e internacionais de padronização, dentre elas destacam-se:

- **ISO/IEC 19794-2:2005** (*Fingerprint Minutiae Data*)
- **ISO/IEC 19794-4:2005** (*Finger Image Data*)
- **ANSI/INCITS 378-2004** (*Finger Minutiae Format for Data Interchange*)
- **ANSI/INCITS 381-2004** (*Finger Image-Based Data Interchange Format*)

O confronto de minúcias em impressões digitais, originário dos estudos de William J. Herschel e Henry Faulds, é a técnica mais utilizada em algoritmos de confronto de *templates* de impressões papilares [Rodrigues e Ribeiro, 2012]. Tal abordagem se assemelha à técnica manual utilizada para a identificação de impressões de dois indivíduos. Um dos métodos mais utilizados é o baseado na técnica desenvolvida por Allan S. Bozorth, que considera através de descritores básicos de cada minúcia, posições x, y, ângulo e tipo, entre minúcias adjacentes. Após considerar e tratar as deformações e distorções provenientes da forma de captura, os conjuntos de descritores constituintes dos *templates* são comparados, sendo considerados compatíveis se os mesmos satisfazem as seguintes condições [Watson *et al*, 2007]:

$$\Delta d(d(P_m), d(G_n)) < T_d \quad (2.1a)$$

$$\Delta \beta(\beta_1(P_m), \beta_1(G_n)) < T_\beta \quad (2.1b)$$

$$\Delta \beta(\beta_2(P_m), \beta_2(G_n)) < T_\beta \quad (2.1c)$$

$$\text{Tipo}(P_m) = \text{Tipo}(G_n) \quad (2.1d)$$

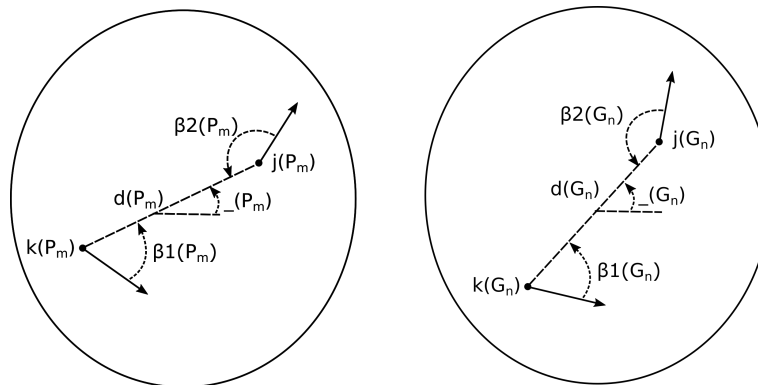


Imagem 2.2 - Pares de minúcias confrontadas, exibindo as propriedades de distância entre as minúcias com as retas projetadas e os ângulos formados entre os pontos marcados [Watson *et al*, 2007].

Na fórmula 2.1a o  $\Delta d$  representa a variação das distâncias interminúcias e  $\Delta \beta$  a variação entre os ângulos das minúcias confrontadas ( $\beta_1$  e  $\beta_2$ ), ambos valores comparados entre dois pares de minúcias nos *templates* questionado e padrão. Os valores  $T_d$  e  $T_\beta$  são os limites

consideráveis (*threshold*) em função da distorção entre as impressões processadas, que podem ser distintas dependendo da forma de captura. Tais valores são comparados com os resultados das funções de cálculo das distâncias euclidianas e das diferenças angulares entre os pontos estudados, P e G.

Outras abordagens utilizam a geração de grafos planares através da triangulação dos pontos característicos mais próximos, gerando um polígono e posteriormente codificando as coordenadas polares de alguns destes pontos, como proposto por Rodrigues *et al*, 2013. Neste trabalho, utilizando uma técnica de triangulação otimizada, método de Delaunay, o algoritmo busca a construção de um vetor de características único e imune a ruídos [Moura, 2006]. Cada ponto característico é processado e gera um subgrafo ligando todas minúcias adjacentes conectadas a ele pela malha da triangulação. O conjunto dos subgrafos e utilizado como vetor de característica da impressão processada e com base neste *array* de vértices são computadas as similaridades entre estes pares, considerando também os parâmetros de distância entre os vértices, a direção em ângulo da minúcia e a diferença entre o ângulo da minúcia processada e a adjacente.

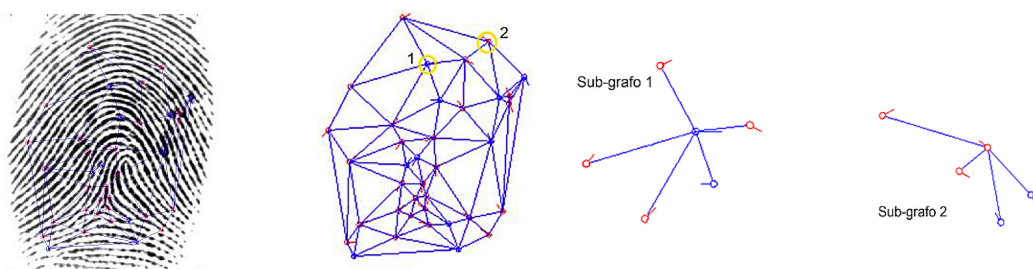


Imagem 2.3 - Minúcias marcadas na impressão digital com o polígono respectivo formado, minúcias 1 e 2 destacadas com os respectivos subgrafos apresentados.

As abordagens mais modernas para extração de características e classificação de impressões digitais utilizam uma variedade de técnicas que consideram elementos dos três níveis citados inicialmente (Global, Local e Detalhe Fino) [Galar *et al*, 2015]:

- **Extração:** Orientação, Pontos Singulares (núcleo, delta), Estrutura das cristas; Respostas aos filtros de imagem (Imagem 2.4);
- **Classificação:** Sintática, Redes neuronais, Confronto de mapas gráficos, Modelos Estruturais (árvores de decisão), Vizinhos próximos, Máquinas de vetor de suporte e outros (Imagem 2.5);

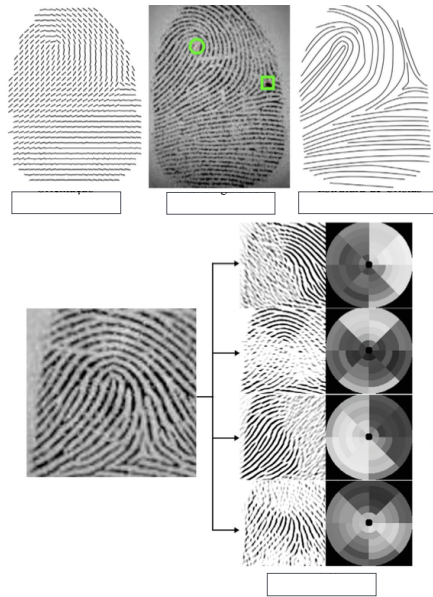


Imagem 2.4 - Apresenta as principais técnicas de extração de características. [Maltoni *et al*, 2005]

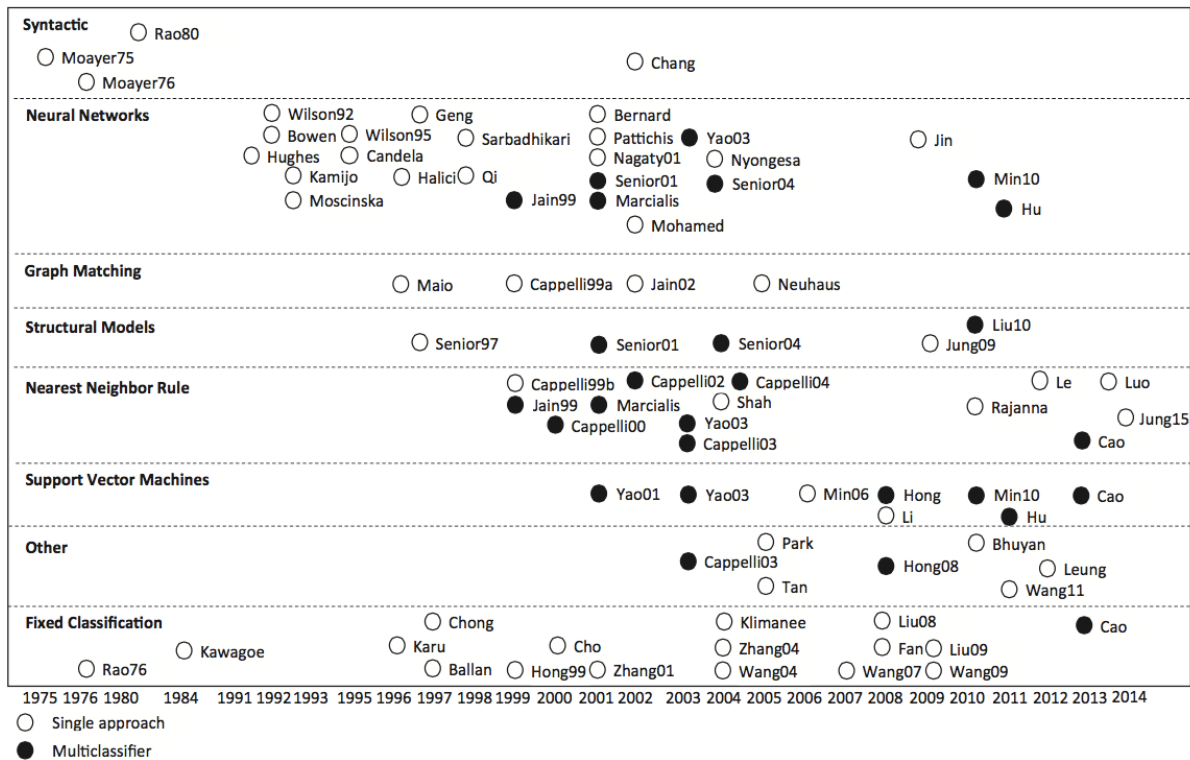


Imagem 2.5 - Apresenta conjunto das técnicas de classificação de impressões digitais [Dorizzi *et al*, 2009].

A comunidade científica publica regularmente inovações e abordagens mais eficientes para a extração de características e para a classificação de impressões digitais, de maneira que constantemente temos métodos alternativos para a elaboração de códigos e aplicações eficientes para executar este trabalho. A classificação especificamente tem importância ímpar por refletir diretamente no desempenho dos processos de identificação, em operações de confronto entre uma impressão e  $n$  impressões distintas. Nesta linha, os AFIS promovem rapidez e

disponibilidade para esta dispendiosa tarefa. Dentre as alternativas disponíveis para implementação de um AFIS existem suítes proprietárias e outras de código aberto, que permitem o estudo e a implementação de soluções de software biométricos utilizáveis em diversos contextos. As suítes de código aberto permitem uma implementação mais flexível e uma adaptação personalizada de uma solução de software para utilização [Usmani *et al*, 2013]. Dentre as alternativas disponíveis e trabalhos publicados, foi estudado um projeto que se apresentou como uma alternativa *opensource*, flexível, leve e eficiente para a criação de uma ferramenta AFIS viável para a utilização nesta solução, o *SourceAFIS*, apresentado mais adiante.

## **2.2. INTEROPERABILIDADE ENTRE AFIS**

As vantagens inerentes ao uso de AFIS estão relacionadas com os benefícios da automação nas ações de identificação de pessoas e também no campo de identificação de impressões latentes [Mayo, 2008]. Em contraste a esta vantagem específica, a crescente utilização dos recursos de automação criou um problema de compatibilidade entre as diferentes soluções adotadas: a interoperabilidade. Por motivos como os comerciais, segredo industrial e proteção de propriedade intelectual, por muito tempo as aplicações produzidas por instituições privadas não contemplaram formas ou protocolos de compatibilidade, vias de compartilhamento entre soluções de diferentes fabricantes.

A interoperabilidade entre sistemas de identificação do tipo demanda a habilidade das aplicações AFIS efetuarem sua comunicação de forma efetiva e eficiente. A adoção de aplicações proprietárias de fabricantes distintos e a falta de uma formatação padrão entre estas soluções acentuaram a limitação na troca de informações onomásticos ou o compartilhamento de dados biométricos entre bases de dados de diferentes entidades governamentais.

Nos Estados Unidos a Academia Nacional de Ciências (*NAS – National Academy of Science*), em seu Comitê de Ciências Forenses, estabeleceu no relatório “*Strengthening Forensic Science in the United States: A Path Forward*” de 2009, dentre oito tópicos de necessidades da comunidade científica forense, como sétimo item é apresentada a avaliação da Interoperabilidade entre sistemas AFIS. O Instituto Nacional de Justiça (*NIJ – National Institute of Justice*), a Associação Internacional para Identificação (*IAI – International Association for Identification*) e o Instituto Nacional de Padrões e Tecnologia (*NIST – National Institute of Standards and Technology*) estruturam grupos de pesquisa para tratar do tema. Os limitadores para efetivar a interoperabilidade entre estes sistemas, segundo a IAI está relacionado aos seguintes pontos [Mayo, 2008]:

- **Vontade política:** a mudança de paradigma causa nos gestores a inércia para a tomada de decisões em projetos de grande porte, frente ao custo, utilização de recursos tecnológicos e humanos adicionais e o impacto nos sistemas legados;
- **Conectividade:** questões técnicas de infraestrutura, segurança e disponibilidade entre estes sistemas deverá ser equalizado entre os pontos a serem conectados, o que necessitará de acordos de cooperação e adequação aos aspectos legais entre os entes participantes.
- **Consolidação:** informações distribuídas, sem um ponto central de localização, podem apresentar inconsistências e limitações para disponibilizar os dados entre os participantes.
- **Precisão:** ao processar milhões de registros biométricos, a solução deve possuir a capacidade de retornar uma lista útil e precisa de candidatos.
- **Gerenciamento de carga:** tratar as demandas e pedidos de alta prioridade oriundos de outros locais remotos onerando da menor maneira a estrutura local. Processos automatizados de gerenciamento e de confronto devem atender a demanda de forma sustentável.

Este conjunto de limitadores segundo Mayo são obstáculos para um modelo funcional da tecnologia, ainda que no campo conceitual. Apesar da interoperabilidade entre AFIS estar em fase avançada de pesquisa nos EUA, seu artigo aponta para os benefícios para o sistema de justiça daquele país caso o sucesso destas iniciativas seja alcançado de forma plena.

Neste campo de pesquisa as entidades norte-americanas NIST e NIJ produziram um conjunto de protocolos e especificações para viabilizar a interoperabilidade entre aplicações AFIS distintos, independentemente de fabricantes. Objetivando habilitar a pesquisa de impressões digitais entre bases de identificações locais ou Estaduais, frente a não existência de sistemas independentes ou protocolos para operações do tipo, os grupos de trabalho estabelecidos para esta tarefa produziram os seguintes documentos de padronização [Chapman, 2013]:

- *Extended Feature Set Profiler Specification (EFS Profiles):* define um conjunto de características a serem utilizadas nas pesquisas de fragmentos de impressões latentes em aplicações AFIS;

- *Markup Instructions for Extended Friction Ridges Features (Markup)*: especifica um conjunto comum de instruções e um roteiro para marcação de características EFS por examinadores.
- *Latent Interoperability Transmission Specification (LITS)*: fornece uma padronização para transações entre agências cooperadas, sejam elas locais, estaduais ou federais. Trata-se de uma especificação em nível de sistema, paralela e compatível com a especificação de transmissão biométrica eletrônica (*EBTS – Eletronic Biometric Transmission Specification*) do Serviço de Informação da Justiça Criminal do FBI (*Federal Bureau of Investigation*).

O protocolo EBTS desenvolvido pelo FBI permite a codificação e a transmissão de dados biométricos entre sistemas interconectados à instituição, baseados no FBI's *Next Generation Identification System*. Fundamentados no padrão ANSI/NIST-ITL, esse conjunto de especificações estabelecem tipos extra e formatos de registros lógicos a serem transmitidos entre os sistemas AFIS da entidade, estendendo as capacidades dos protocolos internacionais para tipos biométricos adicionais, impressões palmares e face, por exemplo [CJIS, 2013].

Já o LITS, de iniciativa do NIST/NIJ, tem o propósito de viabilizar pesquisas independentes, eficientes entre esferas administrativas, em um formato direto, ponto a ponto. A proposta estabelece uma formatação de compartilhamento padrão, simplifica a metodologia de troca de dados entre estas entidades, além de padronizar as notações utilizadas pelos usuários do sistema. Com uma forma mais global o padrão LITS foca na interoperabilidade entre AFIS distintos e independente do fabricante, concebidos para compatibilidade com os padrões *EFS Profiles*, ANSI/NIST-ITIL e no formato FBI EBTS [Chapman, 2013].

Estas iniciativas procuram apontar uma solução para o problema de compartilhamento e interoperabilidade entre sistemas AFIS distintos nos Estados Unidos. Muito focado na realidade inerente ao país, tais propostas atendem as necessidades particulares das instituições americanas e seus sistemas de identificação, entretanto apesar de a proposta considerar a estrutura organizacional entre estados e órgãos distintos, não é apresentada como uma solução aberta, customizável e econômica para localizar e efetuar confrontos biométricos entre sistemas distintos.

A realidade em nosso país remete alguns dos limitadores citados anteriormente e sugerem a necessidade de uma solução semelhante, ajustada às nossas características e peculiaridades. O modelo federativo impõe algumas dificuldades para integrar sistemas

distintos para utilização compartilhada. A adesão voluntária a uma solução de baixo custo e com vantagens mútuas pode ser um incentivo para o compartilhamento de informações estratégicas como dados biométricos.

Nossa proposta aponta para uma solução de código aberto, focado em técnicas e protocolos bem conhecidos pela comunidade de software livre e pela academia. Neste diapasão apresentamos uma forma de viabilizar a operação de sistemas AFIS em uma rede *peer-to-peer*.

### 2.3. APLICAÇÕES PEER-TO-PEER

Sistemas *peer-to-peer* são sistemas distribuídos que provêm o compartilhamento de dados entre nós comunicantes de forma descentralizada. Estes sistemas exercem um número distinto de funções, entretanto sua função primordial é a localização de dados [Gopalakrishnan *et al*, 2004]. Aplicações *peer-to-peer* por natureza não possuem controle centralizado ou organização hierárquica, possuindo simetria nos papéis assumidos pelo nó na rede, atuando simultaneamente como cliente e servidor de recursos. Esta forma de comunicação permite acesso aos ambientes de outros nós e permite acesso da rede aos itens locais [Lua, 2005].

Segundo Androutsellis-Theotokis *et al* (2004):

*“Sistemas peer-to-peer são sistemas distribuídos, consistindo de nós interconectados aptos a se organizarem em topologias de rede, com o propósito de compartilhar conteúdo, ciclos de CPU, armazenamento e largura de banda, capazes de se adaptar a falhas e acomodar populações transferidas entre nós, enquanto mantêm a conectividade e performances aceitáveis, sem necessitar a intermediação ou suporte global de uma autoridade ou servidor centralizado.”<sup>1</sup>*

Em nossos dias protocolos de comunicação *p2p* estão presentes na Internet nos mais diversos serviços, representando montantes entre 60% e 80% do tráfego de dados na rede, em aplicações que disponibilizam arquivos, chamadas de voz, mensagens instantâneas ou mesmo tempo de processamento computacional. Um exemplo é o *BitTorrent*, um dos protocolos de

---

<sup>1</sup>Tradução nossa, considerando o texto original: “Peer-to-peer systems are distributed systems consisting of interconnected nodes able to self-organize into network topologies with the purpose of sharing resources such as content, CPU cycles, storage and bandwidth, capable of adapting to failures and accommodating transient populations of nodes, while maintaining acceptable connectivity and performance, without requiring the intermediation or support of a global centralized server or authority.”.

compartilhamento de arquivos mais populares, que chegou a consumir o montante de 30% de todo o tráfego nas redes IP no ano de 2008 [Spangler, 2015] [Midha *et al*, 2014].

Existem duas classes de redes *p2p* sobre o protocolo IP: Estruturada ou Não Estruturada [Lua *et al*, 2005]. A distinção entre elas resumidamente está relacionada com a rigidez estabelecida em sua topologia e o posicionamento do conteúdo em nós específicos, melhorando a eficiência de consultas subsequentes. As redes estruturadas via de regra utilizam tabelas *hash* distribuídas (*DHT – Distributed Hash Tables*), que são índices de mapeamento e que associam pares de informações com a finalidade de viabilizar uma eficiente forma de localizar recursos.

As vantagens na utilização de uma rede neste formato estão relacionadas às características de tolerância a falhas, independência entre os nós, balanceamento de carga, escalabilidade e principalmente na eficiência em localizar recursos.

Em uma arquitetura distribuída, sem uma organização hierárquica ou controle centralizado, a viabilização do compartilhamento de dados demanda de um formato descentralizado confiável e uma maneira eficiente de localizar os recursos em rede. No caso de *templates* biométricos para esta proposta, viabilizar a localização o prontuário atribuído ao identificador do indivíduo e permitir acesso a estes dados no servidor final.

### **2.3.1. REPUTAÇÃO EM SISTEMAS PEER-TO-PEER**

Em certos ambientes ponto-a-ponto os agentes presentes na rede possuem autonomia para definir seus níveis de participação no sistema, estabelecem a forma de interagir, mediante os requisitos da aplicação, e os meios para atingir os seus objetivos. A operação sustentável de soluções nessa realidade demanda contrapartida de todos os nós, tanto para disponibilizar quanto para utilizar recursos de vizinhos. Este paradigma de comunicação está ligado ao conceito de confiança, que em verdade é uma extensão da nossa realidade, como parte integrante da nossa existência social (Suryanarayana *et al*, 2004).

Reputação segundo Abdul-Rahman *et al* (2000) é a expectativa sobre um comportamento de um indivíduo baseada em informações ou observações sobre seu comportamento no passado.

Sistemas de reputação para sistemas de comunicação são minuciosamente apresentados e discriminados nos trabalhos de Hendrikx *et al* (2015) e Pinyol *et al* (2013). No primeiro citado, é apresentada uma nova taxonomia para sistemas de reputação comparando



pesquisas e sistemas atuais desenvolvidos nesta área. O segundo trabalho apresenta modelos mais cognitivos, detalhando a importância da decisão no processo de confiança.

Genericamente, sistemas de confiança são baseadas em políticas pré-definidas ou em reputação com base na operação do sistema. Tais métricas auxiliam os nós na decisão de efetivar a comunicação com outros nós confiáveis. Para prover esta tarefa, os sistemas de confiança baseados em reputação *peer-to-peer* devem, segundo Koutrouli *et al* (2006):

- **Coletar informações transacionais em cada *peer*.** Nós comunicantes produzem pontuações para a performance de outros nós, informações reunidas localmente como forma de avaliar outros *hosts*. Pontuações individuais ou avaliações constituem recomendações, que são distribuídas nas redes P2P. Cada nó pode armazenar estas informações ou pode disponibiliza-las sob demanda ou via propagação na rede.
- **Produzir valor de reputação.** Reunir informações de confiança que estão relacionados com o comportamento transacional em um nó monitorado, produzindo um valor de reputação para ele. A coleta destas pontuações se torna inviável globalmente ou muito custoso, no caso de redes muito grandes somente parte das transações são coletadas.
- **Ranquear os nós de acordo com sua reputação.** Nós são classificados de acordo com sua pontuação de confiança e comparados entre eles ou com um valor limite definido, permitindo definir se um nó é confiável ou não com base em um limiar (*threshold*) estabelecido pelo sistema.

Suryanarayana *et al* (2004) apresentam uma série de parâmetros utilizados como propriedades que são consideradas na concepção de um sistema de reputação em operações de redes *peer-to-peer*. Dentre os itens listados no trabalho “*A Survey of Trust Management and Resource Discovery Technologies in Peer-to-Peer Applications.*”, destacamos:

- **Controle local:** o controle dos dados do nó pode ser definido pelo nó localmente ou por um sistema de reputação global;
- **Tipo de reputação:** indica o tipo de mecanismo utilizado pelo sistema para pontuação das entidades, definindo a forma de atribuição dos parâmetros de confiança entre os nós do sistema.

- **Custo de largura de banda:** o grau de uso do canal de comunicação para a troca de dados, bem como o *overhead* inserido no sistema em decorrência dos processos de avaliação entre os nós pode ser utilizado.
- **Custo de armazenamento:** representa o montante utilizado para armazenamento dos dados derivados do sistema de reputação. Quanto maior o sistema, maior a demanda para armazenamento destes dados.
- **Confiabilidade:** propriedade do sistema de reputação que avalia o estado passado das entidades participantes no processo de comunicação, se utilizando de estatísticas locais ou recebidas de outros nós para estabelecer níveis de confiança entre *peers*.

Um exemplo de sistema de reputação para rede *peer-to-peer* é o *EigenTrust* [Kamvar *et al*, 2003], um *framework* que permite que entidades decidam quais outros entes são confiáveis para o compartilhamento de arquivos. É um protocolo descentralizado e baseado em DHT. Cada nó mantém um histórico dos demais nós, que são simplesmente a resultante da soma de interações positivas e negativas que derivaram da comunicação entre eles, normalizados entre os valores 0 e 1. Baseando-se nas informações propagadas por todos os nós contatados, uma entidade participante calcula a reputação global para outro nó, considerando a reputação deste nó avaliador. É definido um valor de confiança local,  $S_{ij}$ , como sendo a soma de transações efetuadas pelo *peer i* na interação com o *peer j* [Kamvar *et al*, 2003].

$$S_{ij} = \sum tr_{ij}$$

Os demais nós na rede também armazenarão dados de suas interações com o nó  $j$ , armazenando os números de transações positivas  $sat(i,j)$  e negativas  $unsat(i,j)$  calculando  $S_{ij}$  de forma equivalente desta forma:

$$S_{ij} = sat(i,j) - unsat(i,j)$$

Outros sistemas de reputação são apresentados nos trabalhos acima citados, entretanto o objetivo do trabalho aqui desenvolvido foca no estabelecimento de um sistema de reputação elementar, em um ambiente *p2p* estruturado utilizando tabelas de *hash* distribuídas. Tal implementação segue a motivação semelhante ao estabelecido no desenvolvimento do *EigenTrust*, com a diferença de que a aplicação proposta no trabalho aqui desenvolvido não considera uma reputação global, mas objetiva um sistema de reputação entre os nós

participantes da rede, a fim de estabelecer prioridade no processamento de recursos entre os nós, no caso do AFIS, a execução de confrontos entre as bases de dados biométricas.

### 2.3.2. MÉTRICAS E CUSTOS EM APLICAÇÕES PEER-TO-PEER

Os sistemas *peer-to-peer* implementados sobre o protocolo IP atualmente disponibilizam em seu bojo de funcionalidades, técnicas e protocolos que viabilizam a localização, compartilhamento e transferência de recursos entre seus entes participantes. Além destes serviços, sistemas *peer-to-peer* buscam um satisfatório equilíbrio na demanda por recursos exigidos em ambientes com nós heterogêneos, distintos em capacidades de armazenamento, processamento, conexão ou demanda de seu conteúdo.

Estas necessidades atreladas às características variada dos hosts na Internet podem significar o desbalanceamento da carga entre os nós e limitações no atendimento das solicitações entre eles, seja por excesso de solicitações ou por esgotamento de recursos computacionais. Cada item inserido no sistema possui uma carga associada que pode estar relacionada ao custo de armazenamento, carga no canal de transmissão, processamento ou mesmo memória. A movimentação destes itens entre os nós, resultantes do processo de distribuição e replicação, também impacta na atuação dos *peers*, alterando estas variáveis e influenciando a disponibilidade da rede.

Propostas como as apresentadas em Gopalakrishnan *et al* (2004), Warneke *et al* (2011) e Hsiao *et al* (2011), consideram para redes estruturadas, baseadas na DHT, o gerenciamento de IDs como forma de promover uma equalização entre as cargas dos nós. Avaliando as regiões do escopo estabelecido para os IDs, os algoritmos destas formas de balanceamento definem as faixas de identificadores mais consultados por meio de decisões baseadas em dados coletados no próprio host. As métricas utilizadas neste caso são estatísticas de acesso aos nós vizinhos durante a operação da rede, representando o nível de utilização destes pontos.

Outra maneira de nivelar a carga entre nós é a organização destes *hosts* em grupos, baseando cada agrupamento (*clusters*) segundo seus parâmetros de armazenamento, processamento, memória e latência de rede, como descrito em Ayyasamy *et al* (2010), o exemplo representado por  $w$  a seguir.

Para cada nó  $N_i$ ,  $i = 1, 2, \dots, n$ , seu peso pode ser calculado por:

$$w_i = \frac{\text{Largura de Banda} + \text{CPU} + \text{Memória}}{\text{Latência}}$$

Considerando as variáveis citadas os nós participantes da rede são agregados em grupos “fortes” e “fracos” que recebem conteúdos de dados classificados de acordo com sua popularidade, aqueles mais solicitados são organizados em uma classe específica e replicados para os grupos definidos como “fortes”. No interior de cada grupo são definidos nós líderes que periodicamente recebem informações acerca da carga e da disponibilidade de armazenamento de cada nó. Dentre os nós do grupo, o nó líder verificando que a diferença entre a carga dos nós está acima de um determinado limiar, os dados são redistribuídos.

A abordagem apresentada em Godfrey *et al* (2004) reúne informações de carga dos *peers* relativos aos objetos compartilhados, considerando por exemplo o tamanho do objeto, a popularidade e o tempo de processamento para disponibilidade do mesmo. Na proposta desenvolvida os autores explicitam a utilização de somente uma das variáveis como parâmetro de carga dos nós, o custo de movimentação do objeto, deixando para estudos a consideração de todos os parâmetros juntos. A carga em cada nó, representado por  $l_i$ , é estabelecida como a soma das cargas representadas pelos objetos armazenados neste *host*. Cada nó possui um valor que representa a sua capacidade máxima de carga destes objetos,  $c_i$ . O valor  $c_i$  e o valor de  $l_i$  são as variáveis utilizadas na fórmula de cálculo do fator de utilização de cada nó individualmente e do sistema todo, respectivamente:

$$\mu_i = \frac{l_i}{c_i}, \mu = \frac{\sum l_n}{\sum c_n}.$$

O valor de  $\mu > 1$  atribuído a um determinado nó representa uma sobrecarga do *peer*. Ao passo que todos calculam seus respectivos custos, eles vão sendo definidos como nós “leves” ou “pesados”, dependendo do valor calculado. Estes valores são armazenados em estruturas chamadas diretórios, que possuem IDs conhecidos por todos os outros nós. Servindo como um índice centralizado, estas estruturas armazenam as informações de  $\mu$  que entram em contato durante a operação da rede, monitorando a carga de cada nó conhecido. Caso algum dos nós possua seu valor de carga acima do limite estabelecido, este nó contata o último diretório contatado, provocando a transferência dos dados para nós mais leves.

Algumas das abordagens apresentadas nesta seção foram consideradas para distribuição da carga e para composição do sistema de reputação construído para o sistema apresentado neste trabalho. Na aplicação proposta utilizamos o protocolo Chord como protocolo de localização dos recursos compartilhados, servindo da DHT para distribuição das chaves de mapeamento entre os nós participantes da rede. Esta classe de sistemas foi escolhida pela característica do problema a ser resolvido, que foca em viabilizar a troca de informações

entre Estados mantendo a gestão dos dados compartilhados nos nós de origem. As métricas e custos elencados no texto foram considerados para a construção de um sistema de reputação entre os nós participantes, considerando também técnicas já bem desenvolvidas, tornando a rede sustentável na distribuição de recursos frente as distintas características dos *peers*.

### 2.3.3. PROTOCOLO CHORD

O protocolo Chord em sua concepção é um protocolo *P2P* estruturado, [Lua, 2005] efetua o mapeamento de uma dada chave para um determinado nó de rede. Seu modelo simplifica o design de sistemas e aplicações *P2P* atendendo as necessidades de balanceamento de carga, descentralização, escalabilidade, disponibilidade e flexibilidade na definição das chaves utilizadas [Stoica *et al*, 2001]. Este último se torna atrativo para a aplicação aqui proposta, uma vez que os requisitos de um sistema distribuído em um modelo federativo incluem contar com uma maneira de localizar os recursos baseando-se em dados de identificação dos indivíduos cadastrados. No caso do presente estudo, foi utilizado o número de CPF de cada pessoa identificada.

Em seu cerne, o protocolo Chord provê uma maneira rápida de mapeamento entre as chaves e os nós responsáveis por eles usando o chamado espalhamento consistente (*consistent hashing*), de forma que um mínimo de nós precise mover suas chaves mediante saída de um nó da rede. Com um mínimo de informações sobre seus  $O(\log N)$  nós vizinhos a escalabilidade de uma rede com  $N$  nós é garantida, através da comunicação indireta entre os nós sequenciais.

Na aplicação aqui proposta, cada nó entrante na rede recebe uma chave, ou seja, um *hash* SHA-2 que é produzido a partir da concatenação do FQDN (*Fully Qualified Domain Name*) do nó com a respectiva porta de funcionamento do Chord. Ao passo que outros nós se registram no anel, tais nós recebem chaves semelhantes, baseadas em suas configurações de nome e porta, e posicionando-se conceitualmente em uma sequência circular definida como *identifier circle*, um anel lógico de identificação dentro da extensão de chave de  $m$  bits.

Para cada chave inserida no anel define-se o nó sucessor para esta chave, nó este que receberá e armazenará a entrada, devendo estar posicionado adiante no sentido horário da chave recém-inserida (Imagem 2.6). Esta distribuição tende a balancear a carga na rede com a divisão das chaves para os nós participantes, ainda que de forma imprecisa [Lua, 2005].

Quando um nó entra na rede, assumindo uma posição no anel, parte das chaves armazenadas no nó sucessor devem ser retribuídas ao novo nó, mantendo o espalhamento consistente. Da mesma maneira, após a saída de um nó da rede, as chaves armazenadas neste

nó ausente deverão ser reatribuídas ao nó sucessor. Tal comportamento limita as mudanças de chave de forma localizada, garantindo uma performance  $(\log N)^2$  para entradas e saídas de *peers*.

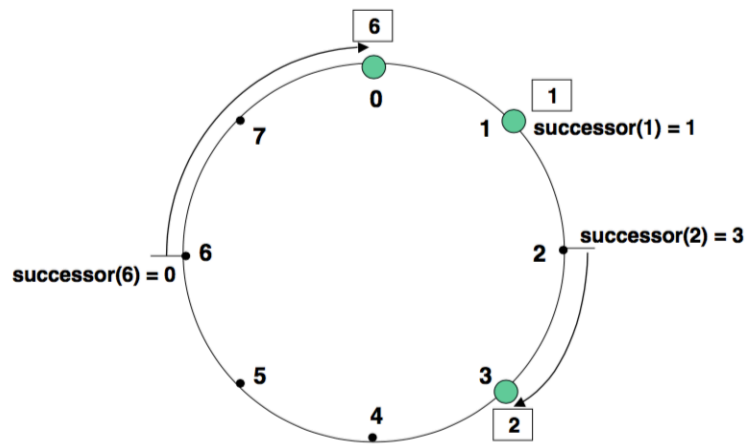


Imagem 2.6 - Anel lógico de identificação com  $m$  igual a 3 bits, estando os nós 0, 1 e 3 inseridos na rede. Neste exemplo a chave 1 é posicionada no nó 1, a chave 2 no nó 3 e a chave 6 no nó 0, sempre buscando o nó sucessor à chave inserida [Stoica *et al*, 2001].

Para viabilizar a pesquisa o protocolo Chord mantém em cada nó participante uma tabela com  $m$  entradas. Esta tabela de índice, *finger table*, armazena os dados de somente alguns outros nós, posicionados a distâncias progressivas a partir do nó origem, seguindo a função  $s = \text{successor}(n + 2^{i-1})$ , sendo  $i$  o  $i$ -ésimo elemento na tabela e  $s$  o nó sucessor para a determinada entrada (Imagem 2.7). Tal comportamento permite ao nó conhecer nós mais próximos do que aqueles mais distantes. Para um dado nó N8 em um anel com  $m = 6$  bits, existem 6 entradas na sua tabela de índices (*finger table*), cada entrada  $i$  aponta para o nó sucessor para a dada chave  $(n + 2^{i-1})$  no anel [Stoica *et al*, 2001].

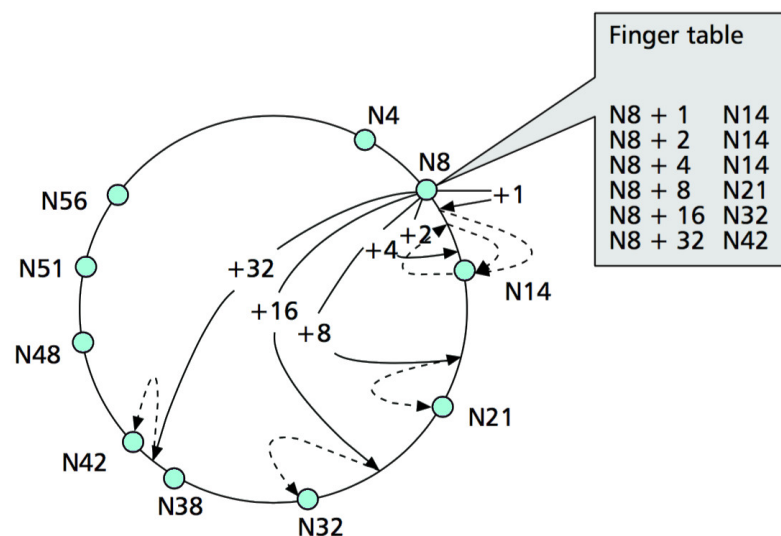


Imagem 2.7 - *Finger table* do nó N8 apresentando os nós sucessores para chaves do anel.

Ao efetuar a busca de para um determinado valor de chave o nó pesquisa em sua *finger table* o sucessor para o valor da chave procurada. Caso exista a entrada na tabela, este será o nó sucessor para a chave em questão. Caso a *finger table* não possua um nó sucessor para a chave pesquisada, é feita então uma solicitação de pesquisa para o nó da tabela mais próximo à chave, pois este nó remoto conhecerá melhor os nós posicionados nesta região do anel. Para redes grandes, com chaves posicionadas em regiões muito distantes a operação recorre nos nós remotos até a localização do nó sucessor da chave pesquisada.

No trabalho aqui apresentado o protocolo Chord permite a distribuição das chaves criadas com base nos IDs (CPF) dos prontuários biométricos presentes nas bases dos nós. A distribuição destes valores na rede permite a construção de um índice de pesquisa descentralizado e compartilhado com todos os pontos do anel. Esta abordagem balanceia o custo de se manter um índice de pesquisa e garante a continuidade da pesquisa em casos falhas nos *peers*.

A integração entre a base biométrica e o sistema aqui proposto passa pelas seguintes operações: Já estabelecido e normalizado o conjunto de *templates* e IDs para o banco local, o módulo AFIS desenvolvido consulta uma visão da base de dados e inserindo os registros retornados em uma rede Chord, processando os *hashes* dos respectivos números de CPFs dos prontuários existentes e propagando estas informações em conjunto com a identificação do nó de origem para os demais nós na rede. Esta propagação distribuirá as chaves pelos diversos nós do anel Chord, permitindo que qualquer outro nó participante possa pesquisar por um CPF específico e localizar o nó proprietário a partir da sua respectiva chave. A proposta é que o protocolo Chord suporte o índice de pesquisa, permitindo maior flexibilidade na operação dos nós independentes, mantendo de forma distribuída a listagem dos CPFs e prontuários presentes no anel. Ao passo que outros nós entram ou saem deste ambiente de comunicação os dados de localização das bases estaduais são propagados para os nós vizinhos, garantindo a manutenção do índice de pesquisa em caso de falha na comunicação ou desconexão.

No caso de pessoas com identidades em vários estados, a ocorrência de uma colisão de *hash* de um mesmo ID em dois ou mais nós, representando um mesmo CPF com prontuários em nós distintos, é tratada criando-se uma lista encadeada a partir do registro já existente na rede. O prontuário inserido passa a ser referenciado pelo registro mais antigo no anel, indicando os nós que possuem prontuários para um mesmo ID.

Estando os IDs propagados pelos nós na rede, a pesquisa de um determinado prontuário se dará da forma como estabelecida no protocolo Chord, através do CPF o nó de origem processa a informação solicitada e busca com auxílio da sua *finger table* o host sucessor ao valor da chave  $k$ , correspondente ao *hash* do CPF pesquisado.

A busca de um nó sucessor dentro da rede é feita consultando a *finger table* armazenada do nó, uma tabela com  $m$  bits de entrada, onde a primeira entrada é o nó vizinho imediato à origem e as demais são nós posicionados no dobro da distância de cada entrada subsequente presente. Este recurso viabiliza uma redução significativa do custo de pesquisa, reduzindo à metade a distância para a chave alvo. Recebendo a resposta da rede, relativa a chave consultada, o nó de origem recebe na resposta a informação de que nó na rede o *template* referente ao CPF pesquisado está armazenado.

Definido o servidor proprietário do prontuário relativo ao CPF pesquisado, o nó de origem aciona o módulo AFIS de comunicação entre servidores, enviando o *template* para ser pesquisado no destino, na base de *templates* do proprietário, nó remoto. A pesquisa executada consistirá em uma operação de verificação entre o *template* enviado e o *template* constatado no destino.

Com uma rede *peer-to-peer* operante, os IDs de todos os prontuários armazenados nos nós participantes ficam disponíveis para serem pesquisados através das chaves distribuídas nos nós, entretanto o acesso ao prontuário dependerá da disponibilidade do servidor proprietário no anel. Devido às características políticas e de propriedade das informações biométricas pertencentes a cada um dos Estados, torna-se mais atrativo para os participantes da rede o processamento local dos confrontos solicitados pelos proprietários dos prontuários, com a propagação posterior somente dos dados compatíveis com o *template* enviado.

Apesar de o estudo aqui apresentado não propor executar a carga direta dos *templates* biométricos para a distribuição destes dados na rede P2P, a aplicação pode ser concebida de forma a permitir anexar tais conjuntos de dados biométricos, permitindo, por exemplo, encaminhar e disponibilizar no anel Chord os *templates* dos usuários cadastrados junto aos índices armazenados nos nós participantes. A decisão de não executar tal carga de *templates* tem como lastro o tempo de convergência em sistemas com bases muito grandes, o volume de dados de *templates* a serem compartilhados poderia comprometer a eficiência da pesquisa em rede diante das limitadas condições de conexão e infraestrutura de muitos órgãos de segurança



dos Estados, especialmente aqueles com menor infraestrutura e orçamento restrito para investimentos na área de tecnologia.

Como aponta Lua *et al*, “*Sistemas DHT consideram que todos os nós participam igualmente no armazenamento de dados compartilhados ou de informações de localização. Isto pode representar um gargalo em nós com baixa capacidade.*”<sup>2</sup>[Lua *et al*, 2005].

Apesar do exposto, a disponibilidade de compartilhamento do próprio conjunto de *templates* junto à chave ficaria disponível como um campo junto à chave  $k$ , se utilizando do recurso de atribuição de *peers* virtuais para hosts com maior capacidade [Hsiao *et al*, 2011], o que agregaria agilidade em confrontos de verificação ou identificação com as mais diversas estratégias. Um detalhe a mais nesta forma de execução é o fato de que o nó que armazena a chave, em conjunto aos *templates*, não tem informação que indique a quem pertence esse conjunto de dados, vez que a chave usada é o resultado do *hash* do ID único, no caso aqui em tela o CPF.

Questões como autenticidade, sigilo e segurança no canal de comunicação não foram tratadas no protótipo e tangenciam o escopo do presente estudo, entretanto podem ser atendidos com protocolos e APIs criptográficas bem conhecidas e padrão no mercado. A distribuição de carga de processamento e pesos nas consultas são questões tratadas nas seções seguintes.

---

2 Tradução nossa, considerando o texto original: “*DHT-based systems assume that all peers equally participate in hosting published data objects or their location information. This would lead to a bottleneck at low-capacity peers*”.

### 3. APLICAÇÃO AFIS

O Projeto *SourceAFIS* disponibiliza uma suíte de software com licença livre, consistindo basicamente em um algoritmo de extração de minúcias com funções específicas para a criação e o confronto de *templates*, na forma de 1 para 1 e de 1 para muitos ( $n$ ). Tal pacote está desenvolvido sob a plataforma “.NET” por meio da linguagem de programação C# e com licença de software livre, BSD.

A utilização de um pacote aberto permitiu o estudo de seu algoritmo e o entendimento das técnicas utilizadas para extração de minúcias, viabilizando o ajuste para a compatibilidade dos dados biométricos disponíveis, personalização da solução e da forma de uso, bem como criação e confronto de *templates* padronizados por entidades como o NIST e ISO. No caso do *SourceAFIS* obtivemos uma biblioteca pronta e disponível para implementação das funções executadas por um sistema AFIS funcional, com funções programadas para a extração de minúcias, criação do *template* e confronto destes *templates*, pelo processo de verificação (um para um) ou identificação (um para 'n').

A API (*Application Programming Interface*) do *SourceAFIS* foi submetida ao programa FVC-onGoing, do Laboratório de Sistemas Biométricos da Universidade de Bolonha, na Espanha, e apresentou boa precisão com índice de 1,17% na taxa EER (*Equal Error Rate*) para o teste padrão. A capacidade nominal da ferramenta é o confronto de 10.000 *templates* por segundo, demorando cerca de 150 milissegundos para extração de cada *template* [Važan, 2012] a partir da imagem *bitmap*, podendo utilizar para confronto *templates* no formato ISO/IEC 19794-2:2011, XML ou binário compacto (formato próprio) [Dorizzi *et al*, 2009].

A extração de *templates* de minúcias nesta biblioteca consiste, de forma resumida, na detecção de pontas de linha e bifurcações na estrutura de cristas das imagens de digitais. O algoritmo do *SourceAFIS* define para cada minúcia encontrada uma coordenada X e Y na imagem, bem como o ângulo de inclinação e o tipo do ponto, armazenando estes dados nas entradas dos formatos citados.

Com base em 8.580 imagens de impressões digitais, disponibilizadas para o estudo pelo Governo do Estado de Rondônia, através do Departamento de Informática e Telecomunicações da Polícia Civil do Estado – DINTEL, foi possível a construção da ferramenta AFIS, bem como a efetivação da análise e entendimento dos formatos utilizados para armazenamento das imagens existentes no banco de identificação local, base utilizada para

confeção das carteiras de identidade para a população desta Unidade Federativa [Santos *et al*, 2015].

Foi verificado que o sistema de registros de identificação em Rondônia armazenava as impressões inscritas no banco utilizando os formatos WSQ (*Wavelet Scalar Quantization*) ou JPEG (*Joint Photographics Experts Group*), com resolução de 96x96 pixels por polegada e dimensões de 404 × 376, salvando cada digital capturada em um arquivo com extensão “.obj.obj”, dentro de uma estrutura de diretórios no computador servidor. Os arquivos armazenados recebiam um nome composto, conjunto do número do prontuário, tipo de impressão e identificador único do arquivo, perfazendo um registro no seguinte formato: “<prontuário>\_<tipo>\_<idObjeto>.obj.obj”. Para fins de exemplificação consideremos o nome “1257577\_fpwlm\_3184874.obj.obj”, neste caso teríamos no arquivo, com base na segunda parte do nome (*fpwlm*), uma impressão digital (*fp*) no formato WSQ (*w – wsq*) do dedo médio esquerdo (*lm – left middle*), atribuído ao indivíduo cadastrado no prontuário “1257577” e com número de identificação única do objeto, atribuído ao arquivo “3184874”.

Ao examinar os dados dos arquivos disponibilizados, verificou-se também que os conteúdos dos arquivos estavam salvos em formato Base64, devido ao sistema utilizado no Estado ser baseado em uma plataforma web.

O sistema desenvolvido, utilizado como módulo AFIS de verificação e identificação foi avaliado com base nesta amostra de dados, objetivou utilizar este conjunto dos 8.580 arquivos para a criação de uma base de *templates* nos formatos *SourceAFIS* binário, ISO e XML, armazenando os resultados em um servidor com o serviço MySQL. Os *templates* persistidos no banco foram utilizados dentro do software desenvolvido para os testes de compartilhamento, localização, verificação e identificação de impressões a serem processadas no sistema. Tal procedimento é detalhado a seguir.

### 3.1.1. CONVERSÃO DE TEMPLATES

A conversão de todas as imagens disponibilizadas foi feita através de uma classe dedicada, controlada por um *Windows Forms* e concebida para listar todos os arquivos dentro dos diretórios e assim processar cada item sequencialmente, extraindo as características de minúcias, criando o respectivo *template* e remetendo ao MySQL um por um. O total de dados processados foi de 621.428.736 bytes, consumindo o tempo de 3 horas e 55 minutos para conversão de todos os 8.580 *templates*. O trabalho foi efetuado por um computador portátil com

CPU 64 bits, Intel Core I7-4650U 2.30GHz com 4GB de memória RAM. Os dados armazenados foram indexados pelo número único do objeto em conjunto com o número do prontuário, disponíveis no nome de cada arquivo. Dentro dos registros inseridos no banco ficaram gravados os seguintes campos de dados: “ItemID”, “Prontuário”, “Template Afis”, “Template Xml”, “Template Iso” e caminho para a imagem “.obj.obj”.

A título de exemplo e comparação entre os formatos guardados, obtivemos para um determinado registro no banco, a partir de uma imagem de 45.909 bytes, o *template* binário do "SourceAFIS" com tamanho de 615 bytes, um *template* XML com tamanho 5813 bytes e um *template* ISO com 630 bytes. Para cada ponto característico encontrado na imagem, o algoritmo define o tipo do ponto e marca sua posição em coordenadas de um plano cartesiano, valores de X e Y, a direção em angulo azimutal, e o tipo de ponto, ponta de linha ou bifurcação. Analisando o *template* XML (Imagem 3.2) é intuitivo verificar as minúcias identificadas e suas características.

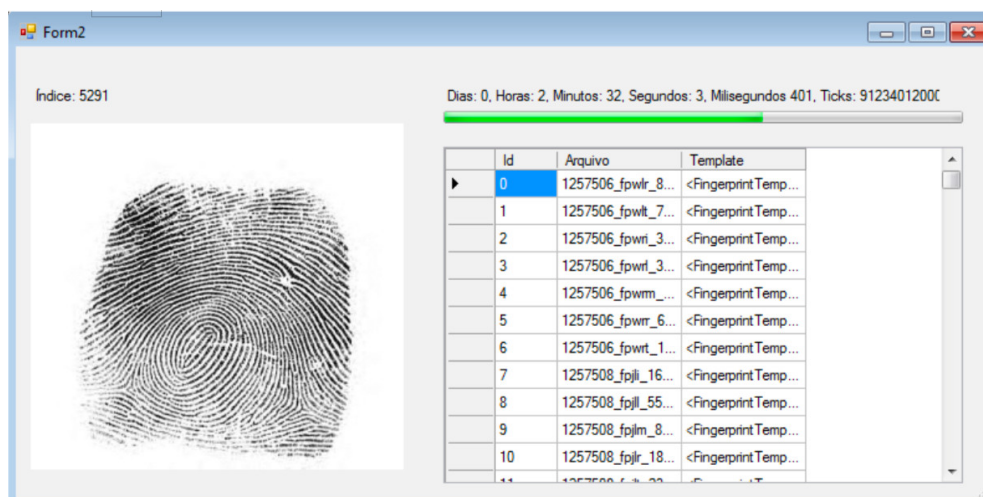


Imagem 3.1 - Janela de status de conversão e inserção dos *templates* no banco.

Ao final do processamento os dados ficam disponíveis no banco para que o módulo de Identificação possa recuperar e utilizar estas entradas para efetuar o confronto na forma 1:n.

```

1. <FingerprintTemplate Version="2" OriginalDpi="500" OriginalWidth="808" OriginalHeight="752">
2.   <Minutia X="411" Y="473" Direction="223" Type="Ending" />
3.   <Minutia X="390" Y="502" Direction="237" Type="Ending" />
4.   <Minutia X="309" Y="138" Direction="119" Type="Ending" />
5.   <Minutia X="247" Y="213" Direction="22" Type="Bifurcation" />
6.   <Minutia X="378" Y="151" Direction="195" Type="Ending" />
7.   <Minutia X="447" Y="361" Direction="200" Type="Bifurcation" /> ... </FingerprintTemplate>

```

Imagem 3.2 - Exemplar de *template* XML criados.

No sistema criado, os dados dos *templates* XML foram utilizados para a visualização dos pontos nas imagens processadas, permitindo a marcação e apresentação do *template* sobre as imagens das digitais, marcando com pontos vermelhos as bifurcações e pontos amarelos as pontas de linhas.

### 3.1.2. MÓDULO DE VERIFICAÇÃO

A API do *SourceAFIS* disponibiliza uma estrutura de classes próprias para o tratamento e organização dos dados a serem processados pela aplicação, abaixo são listadas algumas das classes utilizadas para o processamento das imagens submetidas ao sistema. Parte do código das classes originais foram suprimidos para melhor visualização e entendimento dos conceitos apresentados.

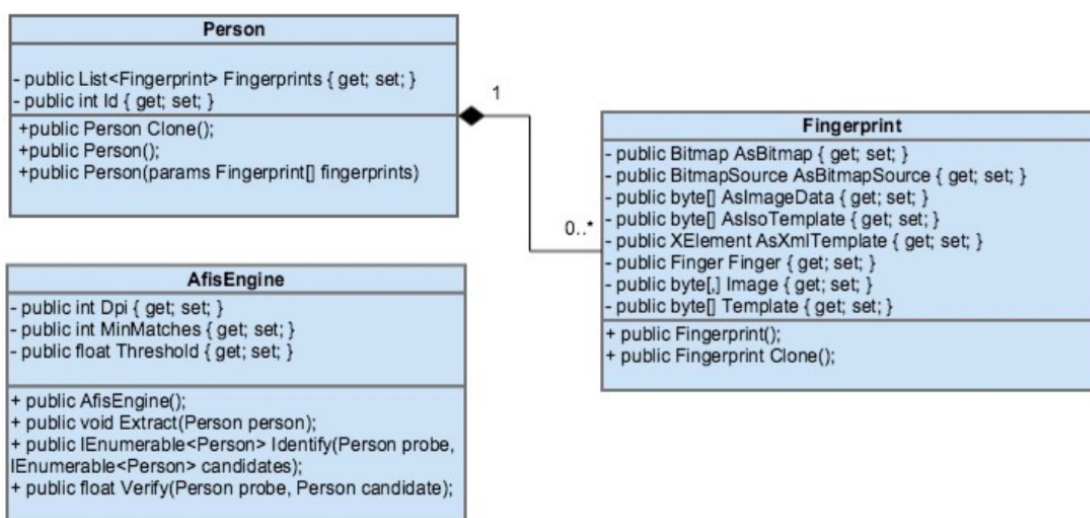


Imagem 3.3 - Conjunto de classes da API do *SourceAFIS*.

O software foi desenvolvido em C#, utilizando a ferramenta IDE “*Microsoft VisualStudio 2012*”. O código implementado acessa os arquivos disponibilizados localmente, lendo seus conteúdos e convertendo as imagens do formato JPG, Base64 e posteriormente do formato WSQ, retornando uma imagem BMP (*bitmap*). Cada impressão, já no formato de imagem *bitmap*, é armazenada em uma lista do tipo “*Fingerprint*” em uma instância da classe “*Person*”. Abaixo se pode observar uma das formas implementadas do processo retrocitado, especificamente para a imagem recebida no formato Base64.

```

//converter String Base64 para byte[]
byte[] imageBytes1 = Convert.FromBase64String(arquivo1);
byte[] imageBytes2 = Convert.FromBase64String(arquivo2);
//Classe para converter WSQ em BMP
WsqrDecoder decoder = new WsqrDecoder();
Bitmap bmp1 = decoder.Decode(imageBytes1);
  
```

```

Bitmap bmp2 = decoder.Decode(imageBytes2);

Fingerprint fp01 = new Fingerprint();
Fingerprint fp02 = new Fingerprint();
fp01.AsBitmap = (Image)bmp1;
fp02.AsBitmap = (Image)bmp2;

Person pessoa01 = new Person();
Person pessoa02 = new Person();
pessoa01.Fingerprints.Add(fp01);
pessoa02.Fingerprints.Add(fp02);

```

Após a inserção dos dados de imagens desejadas, estes objetos da classe “Person” são submetidos ao método *Extract()*, oferecido por uma instância da classe “AfisEngine”, convertendo os objetos na lista “Fingerprints” e preenchendo os demais atributos de cada objeto da classe “Fingerprint”, da forma apresentada a seguir.

```

AfisEngine Afis = new AfisEngine();
Afis.Extract(pessoa01);
Afis.Extract(pessoa02);

```

Ao final da execução das funções do método *Extract()* os objetos “pessoa01” e “pessoa02” recebem os atributos derivados do processamento das imagens *bitmaps* inicialmente submetidas aos objetos “fp01” e “fp02”.

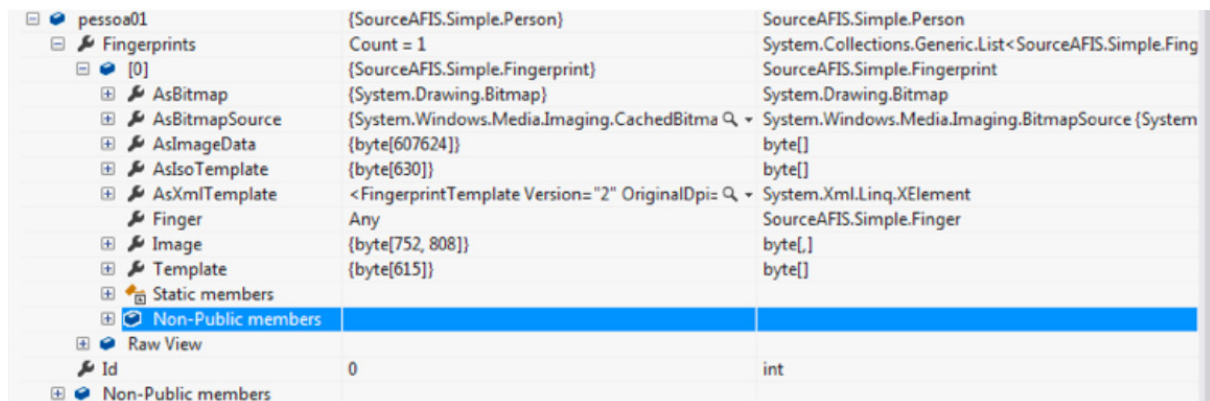


Imagem 3.4 - Janela de variáveis locais do Visual Studio detalhando o objeto “pessoa01” com seus respectivos atributos.

Feita a extração, cada instância “Fingerprint” tem seus atributos definidos, que podem ser utilizados para confronto com outras instâncias da classe “Person” através do método *Verify()* ou podem ser tratados, por exemplo, persistindo os dados no banco, implementado no MySQL. O código apresentado a seguir armazena o *template* XML do objeto “Fingerprint” “fp01” de “pessoa01” em uma *string*, além de verificar se o mesmo é compatível com “fp02” de “pessoa02”, sendo assim feito o confronto 1:1. O retorno do método *Verify()* é um valor numérico onde qualquer valor acima de 0 representa a existência de similaridade entre as

digitais [Važan 2012], quanto maior o valor, maior o nível de similaridade entre os *templates* confrontados. Um valor igual a 0 significa que as impressões são distintas.

```
string templateXml = fp01.AsXmlTemplate.ToString();
float score = Afis.Verify(pessoa01,pessoa02);
string compativel = "";
    if(score > 0)
        compativel = "Sim";
    else
        compativel = "Não";
```

### 3.1.3. MÓDULO DE IDENTIFICAÇÃO

Para o *matching* de 1:n uma lista de objetos do tipo “*Person*” deve ser repassada ao método *Identify()* para que o objeto “*AfisEngine*” execute o confronto entre um objeto “*Person*” e todos os outros objetos da lista, retornando ao final dos testes o melhor colocado, através do método *FirstOrDefault()* na sequência.

No sistema desenvolvido, a lista de candidatos confrontados no método *Identify()* é o conjunto de *templates* já persistidos no banco de dados MySQL, os quais são recuperados e organizados em uma lista do tipo “*Person*”, chamado neste caso de “*database*”, e depois repassada ao método em conjunto com o “*Person*” a ser confrontado.

```
List<Person> database = new List<Person>();

using (MySqlDataReader dr = command.ExecuteReader())
{
    while (dr.Read()) {
        // Para cada registro retornado, insere em database
        int item_id=(int)dr["objID"];
        int person_id=(int)dr["prontuarioID"];
        byte[] template_byte=(byte[])dr["templateBin"];
        String caminho=(String)dr["caminhoImagem"];
        // Método Enroll retorna instância "Person" com as características informadas
        database.Add(Enroll(item_id,person_id,template_byte,caminho));
    }
}

Person matchingCandidate = Afis.Identify(pessoa01, database).FirstOrDefault();
```

Após o confronto, o retorno do método *Identify()* é um objeto da classe “*Person*” com maior score dentre todos os verificados. Cada objeto poderá receber atributos que permitam a sua identificação como nome, identificador ou outro atributo desejado, facilitando a correlação com os dados onomásticos dos registrados no banco.

Na solução implementada, o aplicativo traz como retorno os cinco melhores candidatos, considerando as pontuações de score de um limiar definido, 35 pontos de score. Durante os testes foi perceptível que em alguns casos, quando a imagem submetida para confronto não



possuía minúcias bem definidas ou a imagem apresenta qualidade inferior, ocorria a diminuição de pontos característicos identificados, limitando o tamanho do *template* e aumentando a ocorrência de falsos positivos. Após vários testes com imagens de qualidades diversas, submetendo estas imagens em confrontos 1:1 e 1:n, verificou-se que scores maiores que o valor 35 apresentavam grande correspondência entre as imagens confrontadas, minimizando os casos de falso positivo. Os valores examinados durante os testes ficaram sempre entre os valores 0 e 555, sendo 0 para imagens não similares e 555 para imagens idênticas, quando submetidas cópias por exemplo.

O módulo de verificação e identificação dividem o mesmo *Window Form*, sendo utilizado um *RadioButton* para selecionar a forma como o aplicativo vai operar. Caso esteja marcada a opção “Base Afis”, o aplicativo executa o confronto 1:n. Neste caso sendo a imagem inserida inicialmente confrontada com os *templates* extraídos e armazenados do banco de dados MySQL.

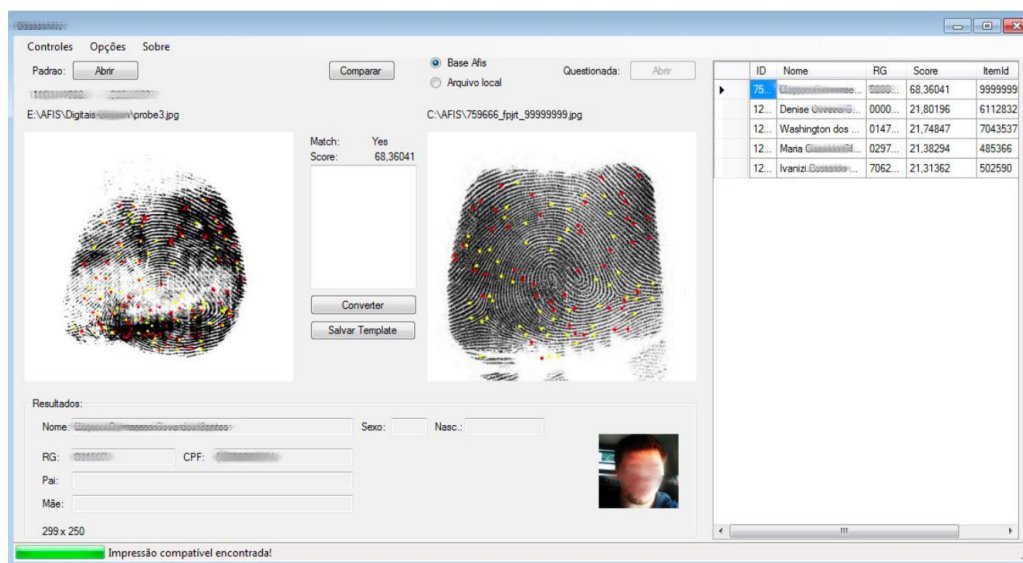


Imagem 3.5 - Tela do módulo de identificação, tendo a imagem de impressão revelada submetida à pesquisa 1:N, retornando o candidato com maior pontuação, positivando a impressão questionada.

Para simular a utilização de fragmentos de impressões coletadas em locais de crime, foi efetuada a coleta de impressões digitais apostas em folhas de papel A4, sendo reveladas com auxílio de pó magnético de cor preta. Após o processo de revelação os papéis foram digitalizados com um equipamento de mesa marca Fujitsu, modelo fi-6140Z, salvando estas capturas em imagens com formato níveis de cinza, com resolução próxima a de 96x96 pixels por polegada. O tempo de extração do *template* da imagem inserida e o confronto com os demais *templates* do banco, já carregados no programa, foi entorno de 100 milissegundos



dependendo da qualidade da imagem inserida, retornando ao final a identidade correta do indivíduo atribuído ao fragmento coletado.

As classes “*Person*” e “*Fingerprint*” suportam serialização, permitindo a criação de *streams* de dados das classes. No processo de recuperação dos dados do banco este recurso foi implementado, visando armazenamento local dos *templates* consultados, possibilitando assim operações *offline*. Os perfis recuperados neste caso são gravados em um arquivo local, chamado “database.dat”. Este arquivo armazena o conteúdo da lista dos objetos “*Person*”, podendo ser recuperado em qualquer momento posterior sem a necessidade de consulta banco de dados, melhorando a eficiência do sistema ao tempo que suprime a fase de consulta durante a operação do software.

```
using (Stream stream = File.Open("database.dat", FileMode.Create))  
    formatter.Serialize(stream, database);
```

O arquivo “database.dat” final alcançou o tamanho de 26,9 MB (28.263.028 bytes), correspondente aos *templates* dos 8.580 arquivos de impressões digitais cadastrados no banco. O tamanho reduzido deste arquivo de *templates* possibilita que o software seja operado de forma *offline*, viabilizando sua utilização em ambientes sem conexão com a Internet ou carregados em equipamentos portáteis, como smartphones e *tablets* por exemplo.

O trabalho desenvolvido resultou em uma solução de software que se mostrou funcional e viável para a conversão e confronto de *templates* de impressões digitais. Os arquivos disponibilizados pelo Governo do Estado de Rondônia possibilitaram a criação de uma base de dados de *templates* para verificação e identificação de impressões papilares, por meio do SDK do projeto *SourceAFIS*.

Os resultados alcançados no desenvolvimento desta solução serviram de arcabouço para a ferramenta AFIS implementada na proposta, módulo para operação das atividades de verificação e identificação de impressões digitais no protótipo AFIS *peer-to-peer*.

## 4. AFIS PEER-TO-PEER

Atualmente cada Estado possui seu próprio órgão de identificação civil e criminal, existe hoje um volume significativo de dados que somente são utilizados localmente, dentro do Estado proprietário de tais conteúdo. Uma forma de compartilhar essas informações biométricas, oferecendo operações de verificação e identificação entre Estados, seria a disponibilidade de um sistema para troca de informações entre os sistemas AFIS dos governos estaduais. Para tanto, a organização destas bases de identificação precisaria atender um formato não hierárquico, alinhando autonomia e garantia de independência de cada ente participativo, mantendo cada qual o seu conjunto de dados em suas próprias estruturas de tecnologia, servidores e ambientes de TI. Desta maneira, o armazenamento dos dados biométricos do Estado estaria sobre a gestão independente de cada instituto.

Nesta seção, são descritas as operações projetadas e desenvolvidas para a interoperação entre diferentes AFIS. O compartilhamento de suas bases de *templates* em um modelo *peer-to-peer*, incluindo a inicialização da rede *Chord*, as trocas de dados de identificação dos nós, o intercâmbio de *templates* de informações biométricas, bem com as operações de verificação de identidades em confrontos 1:1 e 1:n. Ao final são apresentados resultantes da simulação de operação entre nós e considerações sobre a abordagem escolhida, aperfeiçoada durante os testes do software.

### 4.1. PROTOCOLO DE COMUNICAÇÃO

Para o estabelecimento do índice de pesquisa foi escolhido o protocolo *Chord* devido suas características já apresentadas acima. A implementação do protótipo utilizado nos testes foi executada com a adaptação e adição de funcionalidades ao código do projeto *Nchord* [Cencini, 2009], modificado para carga de *templates* e metadados, que disponibiliza as funções *peer-to-peer* para a formação de um anel *Chord* e a propagação de chaves criadas pelo módulo de software desenvolvido neste trabalho.

A comunicação entre os nós se dá por meio de um módulo de comunicação do protótipo AFIS, sobre o protocolo TCP/IP, utilizando as informações de nome do host em conjunto com a porta de operação da aplicação para conexão entre um novo *peer* e um *seed* (nó inicial da rede *Chord*). Estabelece-se um canal de comunicação entre um novo nó e o nó inicial da rede. À medida que outros nós adentram ao anel *Chord*, as chaves de cada nó são propagadas para os demais nós participantes, efetuando a distribuição do índice de pesquisa.

Para a troca dos dados de identificação entre os nós da rede, cria-se um canal de comunicação entre o cliente nó de origem, e o servidor nó de destino. Para tanto, utilizam-se os nomes de domínio configurados em cada nó, associados à porta de comunicação. O estabelecimento de um canal confiável e a participação de um nó na rede de servidores entre os Estados poderia ser definida mediante termo de cooperação entre os entes participantes, por motivos de interesse entre as partes, definindo parâmetros de segurança e procedimentos administrativos, que podem seguir protocolos bem especificados e padrão de mercado. Com o canal de comunicação estabelecido, é efetuado o envio do *template* ao nó proprietário do prontuário a ser pesquisado, obtendo como retorno o *template* compatível ou uma mensagem de não correspondência.

Os formatos utilizados na transmissão de *templates* seguem os padrões ISO, XML e binário compacto, este último nativo do projeto utilizado para confronto de *templates*, a solução *SourceAFIS*, base do protótipo AFIS utilizado, detalhado em Santos *et al* (2015). Poderão ser implementados em trabalhos futuros formatações adicionais como estruturas JSON ou de objetos mais complexos, também baseadas em *Web Services*.

Seguindo a transmissão do *template* encaminhado, uma vez recebido no nó de destino, proprietário do prontuário, o processo AFIS deste nó insere o *template* na fila de processamento correspondente, atribuindo um identificador e um peso à entrada, quando for o caso.

#### **4.2. PROCESSAMENTO DE *TEMPLATES***

Ao estabelecer-se como nó da rede no AFIS *peer-to-peer*, o *host* participante inicia a função interna do nó local processando toda a sua base biométrica, gerando os respectivos *templates* no formato ISO/IEC 19794-2:2011. Caso a base de *templates* já exista no formato em questão, estando o conjunto dos dados normalizado, para cada prontuário cadastrado no sistema é utilizado um ID único associado ao indivíduo, CPF ou título de eleitor por exemplo. No software implementado durante os estudos utilizou-se o número do CPF como ID, a partir de uma *view* do banco de dados de identificação.

Os IDs atribuídos para cada prontuário do banco serão inseridos no índice de pesquisa do *Chord*, por meio do cálculo *hash* do valor do ID, gerando uma chave *k* e sendo distribuído entre os nós do anel para viabilização da pesquisa e localização dos prontuários.

Em cada nó, na medida que as solicitações de verificação ou identificação são recebidas, estes registros e *templates* são persistidos em banco de dados, em uma tabela

utilizada para armazenar os *templates* oriundos da rede, a serem pesquisados. Caso o *template* questionado possua correspondência atestada em algum dos nós participantes em seus sistemas AFIS, este *template* é associado ao prontuário compatível, passando a integrar a base de dados biométrica nos nós de origem e destino.

### 4.3. CONFRONTO DE VERIFICAÇÃO

Conforme mostra a Imagem 4.1, em um processo de verificação de *template* (confronto 1:1), conforme os nós existentes executam pesquisas na rede, ao localizar um nó proprietário para um determinado ID, o nó solicitante estabelece um canal de comunicação TCP com o nó proprietário e envia uma mensagem com o ID a ser confrontado e o *template* de referência (questionado), a ser confrontado nesta base remota.

Após o recebimento correto do *template* no módulo do destino o servidor proprietário insere este *template* na fila de pesquisa de verificação. Na aplicação desenvolvida, a fila consiste em uma tabela identificada com o nome “fila\_ver” no MySQL, SGBD utilizado pelo protótipo. Após persistir tais dados, o *template* questionado é confrontado por meio do algoritmo SourceAFIS [Važan, 2012] com o *template* correspondente no prontuário (padrão), correspondente ao ID especificado na mensagem do nó de origem.

O módulo AFIS do nó proprietário atribui um identificador temporário para esta nova entrada, registra a ocorrência em um log de operações e processa a entrada recém-inserida na chamada fila de verificação. Dentro desta fila de processamento, a ordem de chegada, em conjunto com um peso, detalhado mais adiante, atribuído ao nó solicitante, será utilizada para organização das prioridades na execução dos confrontos. Sequencialmente e sob demanda, o processo AFIS do servidor efetua o processo de verificação de cada entrada na fila de verificação, confrontando os *templates* recebidos (questionados) da rede com os *templates* locais (padrões), disponíveis em sua base de dados.

Após persistir os dados do *template* recebido, o nó proprietário responde ao nó de origem enviando o conjunto de dados relativos ao prontuário compatível, da seguinte maneira: confrontados os *templates*, o servidor proprietário armazena o resultante da operação em um *log* do sistema. Ocorrendo o *matching* (correspondência) no registro processado, o *template* confrontado será então associado ao prontuário do indivíduo ao qual ocorreu a positivação, com uso de uma chave estrangeira criando uma relação para ligação com a tabela de prontuários do nó proprietário. Sendo negativa a correspondência entre os *templates*, o servidor proprietário

descarta o *template* enviado inicialmente e retorna uma mensagem de não correspondência para o nó de origem.

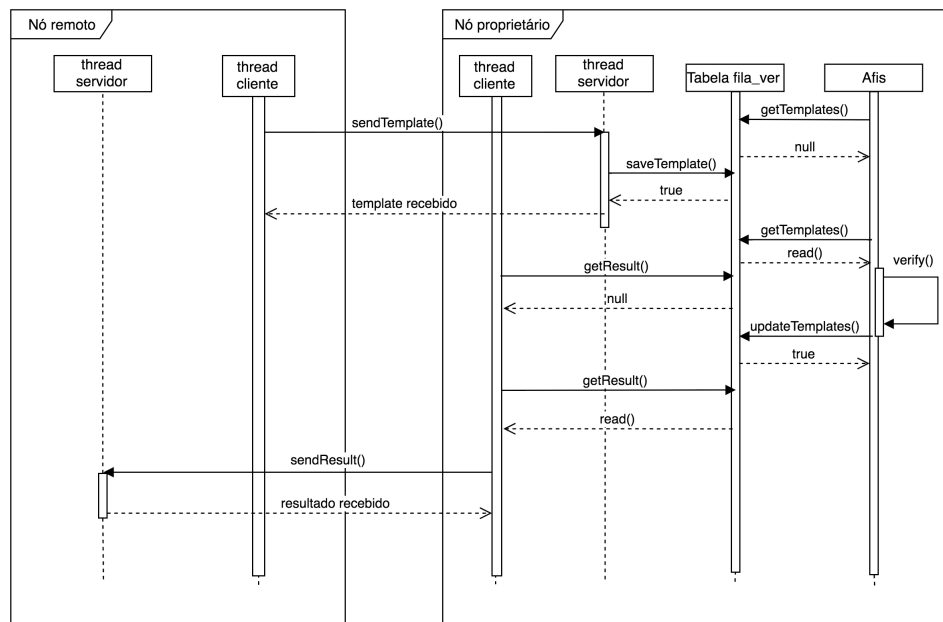


Imagem 4.1 - Sequência lógica de comunicação entre os nós remoto e proprietário na operação de verificação da rede AFIS *peer-to-peer*.

#### 4.4. CONFRONTO DE IDENTIFICAÇÃO

Na ocorrência de um processo de identificação de *template*, aquele em que a pesquisa é feita de 1 para  $n$ , ao invés de pesquisar o ID do *template* a ser confrontado o nó em pesquisa busca o ID de cada nó na rede. Determinando em que nó destino será feita a pesquisa geral (1: $n$ ), é enviada então uma solicitação de pesquisa genérica, constando no ID do prontuário a ser confrontado o identificador do próprio nó destino.

Recebido o *template* da solicitação com sucesso, ao perceber a requisição genérica por meio do uso de seu ID do nó destino, o módulo AFIS atribui um identificador temporário ao *template*, registra o recebimento em um log de operações e armazena esta entrada em uma fila de processamento distinta, chamada fila de identificação “fila\_id”. Nesta fila de processamento, o conjunto do peso do nó solicitante e a ordem de chegada do item na fila é o que determina a sequência de processamento e de submissão do *template* à pesquisa 1 para  $n$ . De forma escalonada e assíncrona, o processo AFIS segue a fila de prioridades buscando em sua base de *templates* aquele compatível com o registro testado. Após o envio do *template* por parte do nó de origem, o servidor proprietário responde à origem com uma mensagem de sucesso no recebimento do arquivo e encerra a conexão.

O *template* enviado, presente na fila de identificação, aguarda o processamento das entradas com maior prioridade e na sequência é submetida ao confronto 1:n, após a conclusão do processamento desta entrada, o módulo de comunicação do AFIS acessa o nó de remetente do *template* testado e retorna o resultado da pesquisa. Caso o resultado do confronto de identificação retorne *matching* com um *template* compatível, o nó proprietário associa o *template* testado com uma chave estrangeira ao prontuário correspondente e envia na resposta ao nó de origem com os dados deste prontuário, garantindo assim a identificação do *template* testado. Caso o resultado seja negativo, o nó proprietário descarta o *template* processado e retorna uma mensagem de não correspondência para o nó de origem.

A sequência lógica de operação para o caso do processo de identificação é semelhante ao de verificação, com as mesmas duas *threads* cliente e servidor em cada nó participante. Entretanto, é utilizada uma tabela distinta como fila de identificação, definida como “fila\_id” no MySQL, e também o processo de confronto de *templates* executa a operação da forma 1:n, confrontando o *template* questionado com os demais presentes no banco de *templates* do no local por meio do método *Identify()*.

Todo o trabalho de processamento e confronto de *templates* foi efetuado utilizando uma solução de software baseada no projeto *SourceAFIS*, com o algoritmo adaptado para trabalhar com *templates* de teste, amostra da base de prontuários biométricos da Secretaria de Segurança do Estado de Rondônia, sistema apresentado em [Santos *et al*, 2015].

#### **4.5. PRIORIDADE E BALANCEAMENTO DE CARGA NOS NÓS**

Em uma aplicação real entre nós representando Estados, os investimentos afetos aos processos de automação e informatização das atividades de identificação tendem a acompanhar o volume de informações que estes órgãos tratam em seus trabalhos correntes, de forma proporcional ao número de habitantes. Estados maiores, com maior população, demandarão maiores investimentos e melhor infraestrutura para o tratamento e processamento de dados biométricos do que aqueles com menor número populacional.

Neste diapasão, é previsível que na aplicação proposta existirão, entre os nós participantes, divergências quanto as capacidades de processamento e memória [Karger *et al*, 2004], assim como diferentes níveis de robustez, significando que distribuir igualmente a responsabilidade não é necessariamente a melhor opção [Epsztejn *et al.*, 2006]. Faz-se necessária então a definição de um formato justo de balanceamento nas atividades de

verificação e identificação dos *templates* questionados junto às bases de identificação dos nós da rede Chord.

Dentre os parâmetros levantados nos *peers* da rede, com objetivo manter uma justa distribuição do processamento dos *templates*, abordagens como a apresentada em Garofalakis et al. (2009) e Chawachat et al. (2014) definem estratégias para construção de métricas e custos para organização das filas de identificação, semelhante a algumas apresentadas em Mondal et al, (2003). No caso aqui estudado é considerado por definição, índice de distribuição de carga, disponibilidade para requisições, custo para armazenar os dados, etc.

A forma de organização das filas de processamento dos *templates* deve estabelecer parâmetros de priorização de solicitações, de maneira a estimular os nós da rede a disponibilizar seus recursos computacionais e dados biométricos.

As filas de verificação e de identificação são tabelas utilizadas para organizar a chegada das requisições de consultas dos nós participantes da rede. Estes registros objetivam a organização e o escalonamento justo do processo de confronto entre os *templates* questionados e os padrões do banco pertencente ao nó proprietário. Neste cenário alguns nós serão mais requisitados que outros, por exemplo, um estado como São Paulo receberá mais requisições de confrontos do que um estado como Rondônia. Torna-se imperativo um ajuste nas prioridades de processamentos para alguns casos, considerando suas capacidades de processamento, número de requisições, volume de *templates* encaminhados, número de registros no banco, etc.

Esta proposta define que, conforme as requisições chegam no módulo de transmissão do AFIS, o registro recebido entra na fila correspondente ao tipo de confronto requisitado, verificação ou identificação, recebendo inicialmente o número da sua posição nela, ordem de chegada. Durante o período de operação do nó na rede, a ocorrência de consultas e interações com os demais entes participantes permite ao nó proprietário coletar informações associadas aos outros nós, viabilizando a construção de pesos para as requisições e para as operações de confronto biométrico.

O sistema proposto utiliza um algoritmo de escalonamento de múltiplas filas, o processo dentro do AFIS implementado mantém organizada uma fila de verificação e outra de identificação. O processamento da fila de verificação representa um custo computacional menor para o sistema do que o processamento da fila de identificação. Devido à necessidade de resposta imediata para o nó de origem, esta fila é selecionada sempre que existirem entradas a serem processadas, com prioridade maior do que a fila de identificação. Dentro desta fila a

ordem de chegada é definida como a ordem de processamento, seguindo o conceito de fila, FIFO (*first in, first out*).

Sabendo que o custo para o processamento da fila de identificação é maior devido à natureza da pesquisa de 1 para n, a organização da sequência e o processamento das entradas neste caso é mais dispendiosa e deve considerar outras métricas, a fim de favorecer aqueles que atendem e respondem melhor a rede. As seguintes variáveis são consideradas para o cálculo de prioridade:

- Ordem de chegada (OC): na sequência da ordem de chegada é atribuído um valor crescente ao item recebido para processamento;
- Disponibilidade do nó (DN): em cada tentativa de consulta de verificação ou identificação em outros nós, a estes são atribuídas pontuações, cumulativas.
- Número de requisições do nó (NQ): cada requisição representa um custo de operação que é atribuído ao nó solicitante, este valor também é cumulativo;
- Número de respostas do nó (NR): em consultas a outros nós na rede, as respostas fornecidas por estes nós constituem pontuação para o nó consultado. Valor cumulativo.
- Número de prontuários disponíveis no nó (NP): o número de itens em sua base sendo disponibilizado para consulta representa pontuação para o nó participante. Valor proporcional ao tamanho da base.

Através da fórmula apresentada em (1), calcula-se a prioridade para cada item da tabela, considerando nesta função os demais pesos, quando disponíveis. O menor o valor de custo representará maior prioridade de processamento.

$$Custo: OC - \left(\frac{1}{2} * DN\right) - \left(\frac{1}{2} * NR\right) + \left(\frac{1}{10^2} * NQ\right) - \left(\frac{5}{10^6} * NP\right) \quad (1)$$

A fórmula proposta foi concebida a partir da análise das grandezas utilizadas como variáveis de controle na função objetivo, tendo como foco definir um custo menor para os nós que disponibilizam mais seus recursos na rede. A definição dos coeficientes na equação foi feita considerando a proporção entre a Ordem de Chegada e as demais variáveis, a razão entre o número de requisições, disponibilidade, consultas e tamanho da base de dados foram ajustadas para influenciar proporcionalmente o resultado final do custo. Cada variável na fórmula linear possui uma constante que representa o peso desta variável na função.



Foi avaliado o número de *templates* a serem processados nos nós participantes em uma aplicação real. No caso do Estado de São Paulo, o maior em números populacionais segundo dados oficiais da Secretaria de Segurança Pública do Estado no ano de 2015, foram cadastrados cerca de 20 milhões de pessoas, cerca de 48% da população do Estado que é de 41.262.199 segundo dados projetados para o ano pelo IBGE [SESP, 2014][IBGE, 2010].

Uma base de prontuários com esse volume de dados proporcionalmente terá uma carga maior de solicitações a serem atendida aos nós da rede, o que demanda um custo mais elevado para o processamento destas requisições. Ao se exigir uma maior infraestrutura para suportar as operações no sistema AFIS, define-se um menor custo para o processamento de solicitações do nó São Paulo, definida com influência do tamanho se sua base de *templates*. São considerados também obviamente a ordem de chegada (OC) com um fator de peso maior, seguido da disponibilidade do nó (DN), número de respostas (NR), como fatores que diminuem o custo. O número de requisições (NQ) também é utilizado, mas para incrementar o custo com peso relativo menor, é necessário de um alto índice de requisições do nó para alterar influenciar o aumento do custo.

Didaticamente, considerando a ordem de chegada como custo inicial, ficou estabelecido que, a cada 2 pontos de Disponibilidade do Nó 1 ponto é subtraído do custo. Da mesma forma, a cada 2 pontos de Número de Respostas atribuída ao nó mais 1 ponto é subtraído do custo. Na sequência, 100 pontos no Número de Requisições do Nó representarão um ponto a mais na Ordem de chegada, aumentando o custo nesse caso. Ao final, considerando o número de prontuários disponibilizados pelo nó avaliado, a cada um milhão de prontuários disponibilizados um ponto vai ser subtraído do custo. Obviamente que os valores inteiros apresentados foram utilizados a fim de facilitar a ilustração, mas na operação da fórmula em questão, o resultado do custo para cada valor inserido será considerado, ainda que num formato decimal.

Tais valores coeficientes foram selecionados após ciclos de testes com simulações de interações entre nós, randomizando os valores DN, NQ, e NR, em condições semelhantes ao de operação real, avaliando-se sempre o rearranjo das posições das requisições dentro desta fila de prioridades. Após testes e ajustes, ao perceber um escalonamento justo das entradas processadas, baseando-se nos princípios acima definidos e de forma empírica, a fórmula foi escolhida.

A partir da prioridade resultado da função formulada em (1), a lista de identificação é definida e cada entrada na lista é processada, submetendo o *template* encaminhado pelo nó de origem ao confronto com todos os outros *templates* do banco local.

A aplicação proposta foi construída seguindo os passos desenvolvidos e apresentados na seção 2.4. A solução concebida foi programada na linguagem C#, sobre o *framework* .NET versão 4.5, base de dados implementada em MySQL e com recursos de comunicação em rede desenvolvidos com as bibliotecas “.NET Remoting” e “.NET Sockets”. A escolha deste conjunto foi feita levando em consideração a facilidade de implementação, familiaridade com a tecnologia e compatibilidade com os projetos utilizados como base do trabalho, soluções *Nchord* e *SourceAFIS*.

**Tabela 4.1 - Fila De Identificação Simulada, Ordenadas Pelo Valor Do Custo.**

| Nó de Origem | Ordem | Custo  | DN  | NQ  | NR  | NP       |
|--------------|-------|--------|-----|-----|-----|----------|
| PA           | 11    | -57,87 | 634 | 701 | 523 | 3638904  |
| MS           | 17    | -52,20 | 601 | 492 | 475 | 1175532  |
| PR           | 13    | -52,06 | 547 | 744 | 485 | 5013372  |
| AL           | 2     | -51,72 | 519 | 77  | 506 | 1497837  |
| RR           | 6     | -49,37 | 598 | 289 | 405 | 216230   |
| MA           | 23    | -44,34 | 509 | 185 | 396 | 3155899  |
| RS           | 20    | -43,09 | 539 | 673 | 325 | 5133086  |
| RO           | 16    | -36,87 | 451 | 50  | 312 | 749956   |
| DF           | 1     | -33,96 | 454 | 305 | 221 | 1233677  |
| PE           | 19    | -32,40 | 390 | 359 | 261 | 4222295  |
| CE           | 8     | -32,08 | 530 | 351 | 94  | 4057143  |
| BA           | 12    | -31,12 | 331 | 299 | 254 | 6728115  |
| AC           | 7     | -30,64 | 470 | 291 | 159 | 352108   |
| PI           | 21    | -28,71 | 484 | 43  | 118 | 1496813  |
| RJ           | 24    | -26,32 | 366 | 314 | 138 | 7675166  |
| MT           | 18    | -25,82 | 508 | 361 | 37  | 1456859  |
| SP           | 22    | -21,25 | 187 | 599 | 96  | 19805856 |
| TO           | 14    | -20,11 | 283 | 726 | 155 | 664054   |
| PB           | 26    | -19,79 | 321 | 562 | 120 | 1807933  |
| ES           | 10    | -19,39 | 238 | 302 | 159 | 1687177  |

|    |    |        |     |     |     |         |
|----|----|--------|-----|-----|-----|---------|
| MG | 15 | -13,38 | 134 | 570 | 81  | 9406718 |
| SC | 5  | -11,48 | 148 | 373 | 69  | 2999249 |
| AP | 25 | -10,92 | 159 | 344 | 113 | 321372  |
| RN | 3  | -8,55  | 105 | 360 | 64  | 1520653 |
| AM | 9  | -4,10  | 58  | 241 | 30  | 1672313 |
| GO | 4  | -4,00  | 40  | 745 | 34  | 2881818 |
| SE | 27 | -3,85  | 77  | 642 | 57  | 992648  |

Pesos como poder de processamento ou capacidade de memória poderiam ser utilizados também para definir a prioridade na fila de identificação, ajustando a prioridade da forma que a aplicação corresponda à necessidade da rede.

A “Tabela 2” apresentada é resultante de uma simulação, inserção de 27 entradas na fila de identificação da aplicação proposta, representando solicitações de confronto dentro do sistema AFIS, sendo uma solicitação para cada um dos nós, representando estados brasileiros. Para fins de teste o valor relativo ao número de prontuários de cada nó foi definido como a parcela de 48% do montante populacional de cada Estado, seguindo a mesma taxa de cadastramento biométrico do Estado de São Paulo e considerando dados populacionais dos estados segundo censo do IBGE [IBGE, 2010].

Para cada estado, considerando seus tamanhos de bases de dados, foram produzidos valores aleatórios para a disponibilidade do nó (DN) e para o número de requisições (NQ). O número de respostas (NR) foi definido dentro da faixa do valor de disponibilidade do nó (DN), representando as respostas recebidas das consultas.

Analisando os valores resultados da simulação podemos observar que na forma aqui proposta, somente a Ordem de Chegada não é suficiente para definir a prioridade de processamento, nós com grandes bases de prontuários e *templates* biométricos terão prioridades de processamento, influenciados em grande parte pela sua ativa participação na rede. É o caso das entradas dos estados DF e MS, em que a entrada do MS recebeu melhor custo apesar de ter entrada depois do registro atribuído ao DF na fila de processamento. Com melhores valores de disponibilidade e mais que o dobro do número de respostas à rede, a entrada do MS é privilegiada com o cálculo de custo considerando também estas variáveis.

É possível observar na simulação que o tamanho da base disponibilizada, associado com a participação ativa do nó na rede, são fatores que melhoram a posição das entradas de

processamento e garantem uma organização justa de posicionamento de entradas de confronto na fila AFIS de cada nó.

Estudando os dados da “Tabela 2” evidenciamos a influência dos fatores de peso do número de respostas elencados na fórmula no reordenamento da fila, visível no gráfico comparativo entre Custo e Número de Respostas, apresentado a seguir.

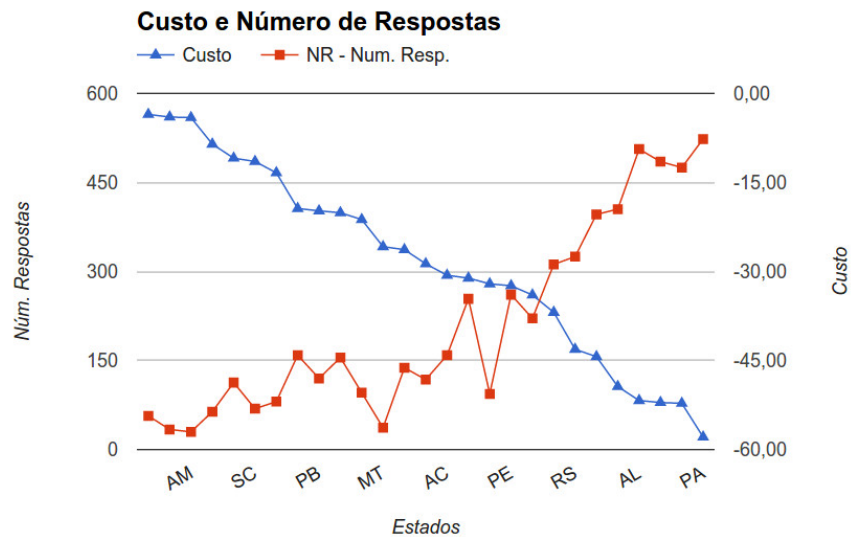


Imagem 4.2 - Regressão do valor do custo frente ao número de respostas recebidas de cada nó.

A Imagem 4.2 representa o comparativo entre o valor do número de respostas e o custo calculado para cada nó. A proposta define a utilização do custo como forma de escalonamento da fila de identificação, neste caso os menores custos seriam processados primeiro. Constatou-se pela simulação uma relação forte entre a quantidade de respostas de cada nó e seu respectivo custo calculado, pois quanto maior o número de respostas do nó, menor o seu custo.

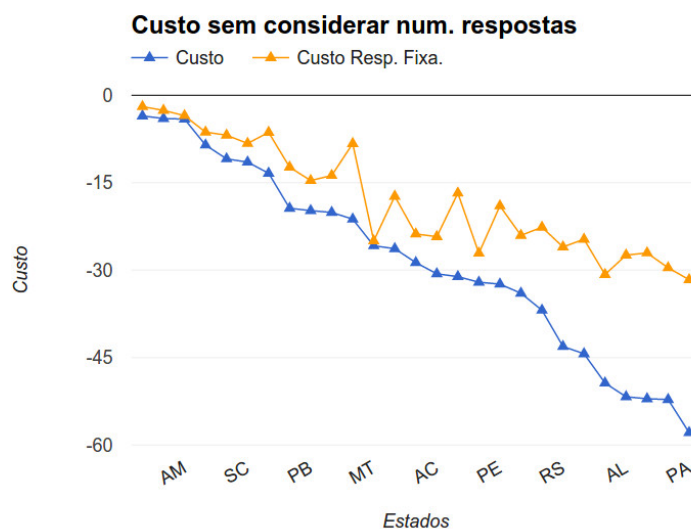


Imagem 4.3 - Projeção comparativa entre o custo padrão e o custo calculado sem considerar o Número de Respostas – NR.

A imagem 4.3 apresenta o custo calculado para cada nó, comparando com um custo recalculado sem a utilização do parâmetro Número de Respostas – NR. Neste caso observamos que as entradas oriundas de nós com valores NR maiores sofrem um ajuste maior no valor do custo, resultando num reposicionamento mais acentuado na fila de prioridade.

A relação entre o custo e a Disponibilidade do Nó é apresentado na Imagem 4.4, representando o valor de custo para as entradas, frente projeção do valor DN de cada um destes itens.

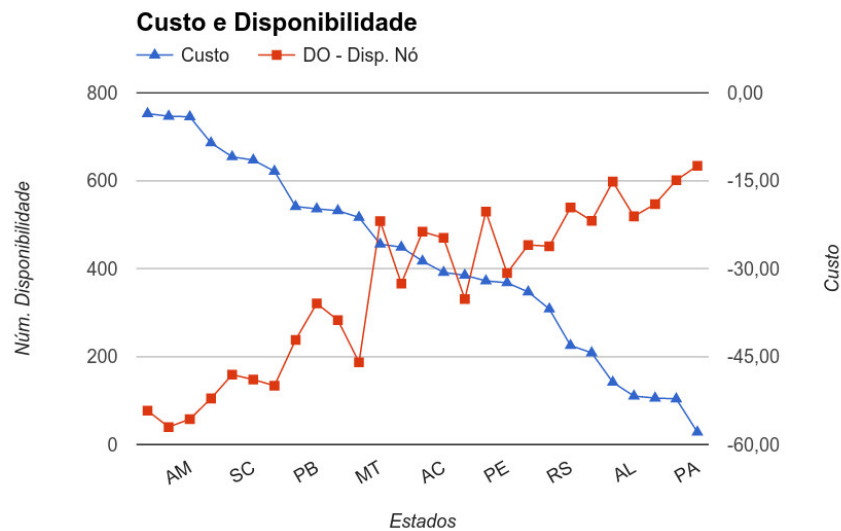


Imagem 4.4 - Regressão do valor do custo frente aos valores de disponibilidade (DN) de cada nó.

Percebemos que, de forma similar ao parâmetro Número de Respostas (NR), o parâmetro Disponibilidade do nó influencia o custo final de forma significativa, refletindo também no reordenamento da fila de prioridades. Abaixo é apresentado o comparativo entre o custo e o custo recalculado desconsiderando o parâmetro NR.

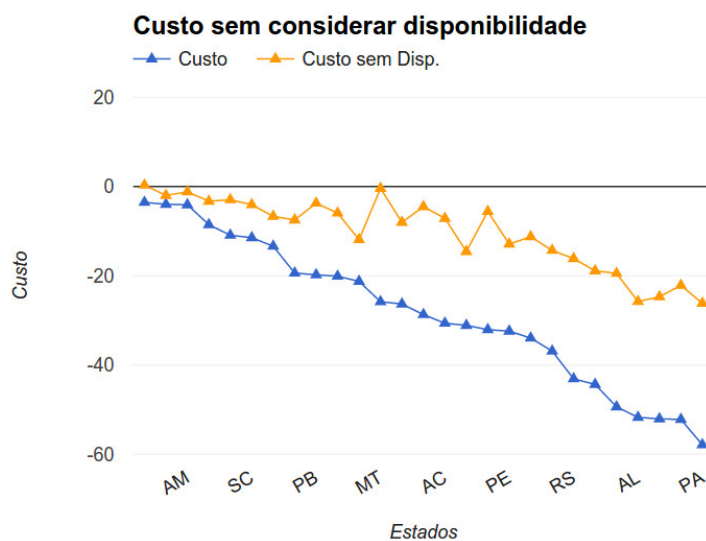


Imagem 4.5 - Valores de custo para cada entrada da tabela frente aos valores de Disponibilidade do Nó – DN.

Similarmente ao considerar a Ordem de Chegada de todas as entradas, verificamos que a influência deste parâmetro no cálculo de custo é considerada, porém de forma mínima, vez que o objetivo é controlar a prioridade da fila baseado nos fatores de peso escolhidos para o sistema.

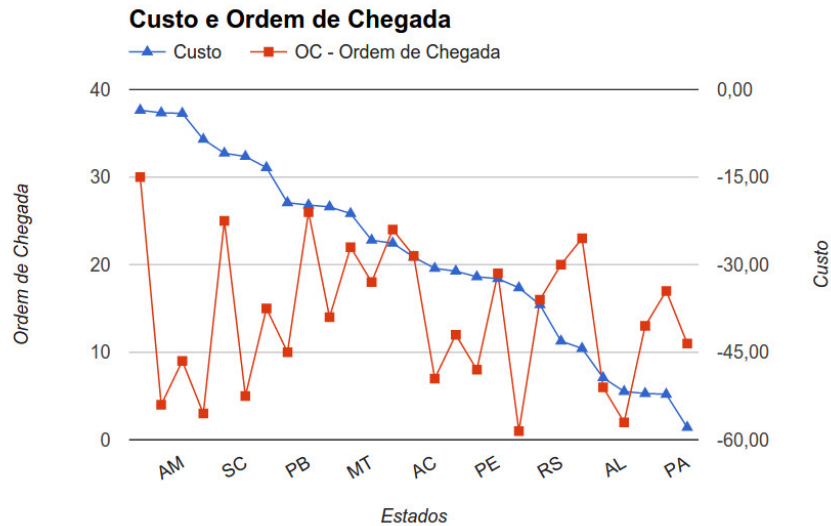


Imagem 4.6 - Gráfico ilustra o valor do custo frente a variação na Ordem de Chegada de cada entrada. Percebe-se pela distribuição que grande parte das entradas têm sua posição redefinida se comparada a ordem de chegada inicial.

Efetuada a projeção do custo sem considerar no cálculo a ordem de chegada (Imagem 4.7), percebe-se a baixa influência do custo frente a outros valores. O valor da ordem de chegada representará fator determinante se as demais variáveis das entradas processadas forem semelhantes ou próximas.

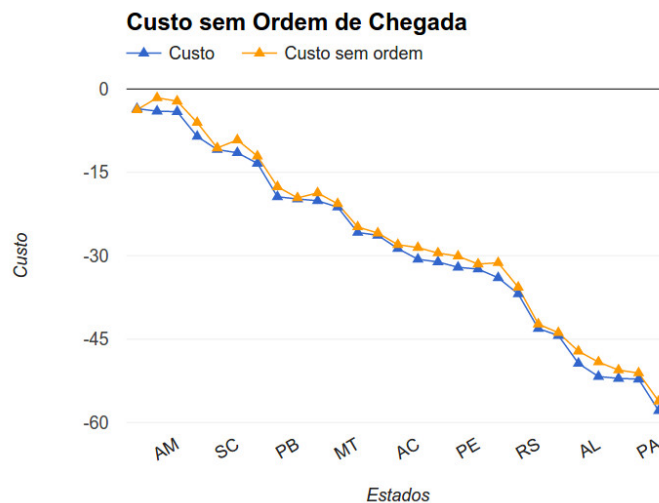


Imagem 4.7 - Plotagem do custo frente ao custo recalculado, sem considerar na fórmula o valor da Ordem de Chegada – OC.

De forma semelhante, o Número de Requisições não alterou de forma significativa a posição das entradas na fila de processamento. Tal fato se justifica no coeficiente escolhido

para esta variável, que altera significativamente o custo mediante um alto índice de requisições de um nó, evitando inundações de requisições dos nós participantes.

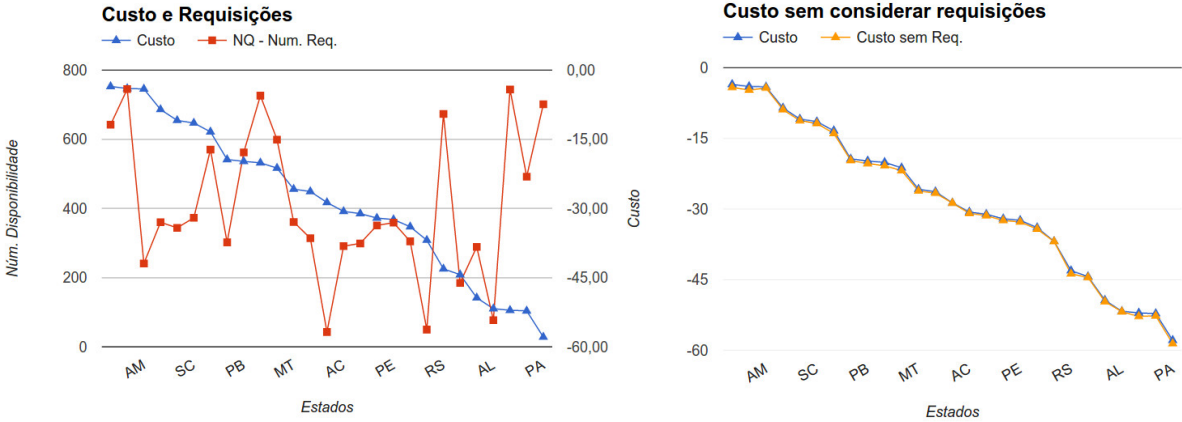


Imagem 4.8 - Representação do custo frente aos dados de Número de Requisições (NQ) de cada entrada. Também a variação do cálculo de custo normal e o custo sem considerar a variável NQ.

Finalizando a análise dos valores, verificamos o fator de influência do número de prontuários de cada dó mediante o cálculo do Custo para cada entrada na fila (imagem 4.8). Neste caso fica fácil perceber que nós com bases de dados maiores terão vantagens pelo cálculo do custo, representando um reposicionamento mais expressivo e um maior privilégio no processamento de suas requisições da rede.

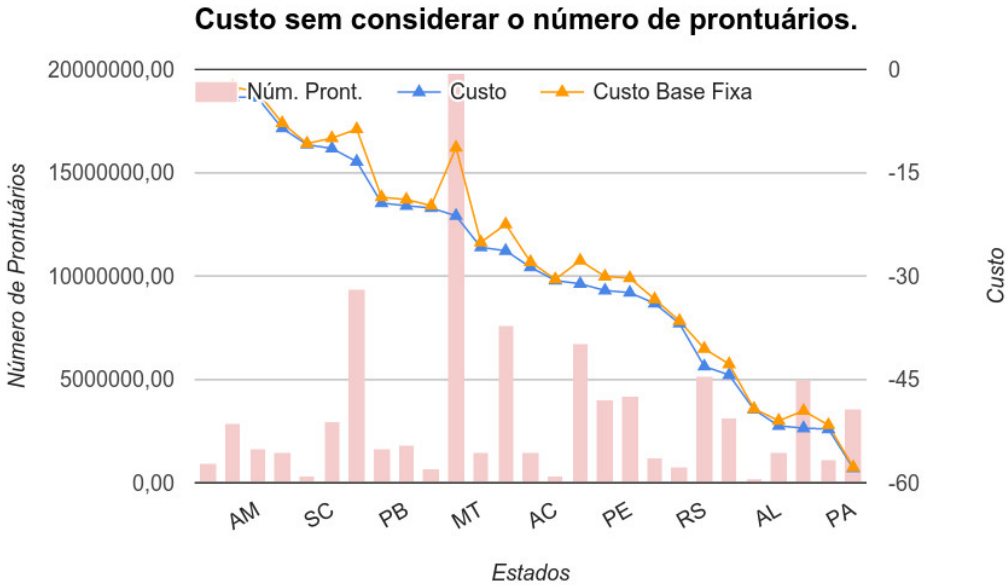


Imagem 4.9 - Custos calculados para a fórmula proposta frente ao recálculo do custo sem considerar o tamanho da Base de Dados. O gráfico de barras contrasta com os demais pontos, apresentando o tamanho da base de cada nó da entrada na fila.

Os resultados apresentados pela simulação definem um comportamento esperado pelo objetivo da função construída, um escalonamento definido pelas variáveis OC, DN, NQ, NR, e NP. A função concebida mostrou-se satisfatória para o cálculo de custo para cada entrada na fila de confronto de Identificação, permitindo a utilização das variáveis atribuídas aos nós participantes para alterar a ordem de chegada de cada entrada na fila. Na próxima seção apresentamos os resultados práticos desta proposta, utilizando as soluções desenvolvidas ao longo do trabalho em uma aplicação real, para compartilhamento e confronto de *templates* biométricos em uma rede *peer-to-peer*.



## 5. RESULTADOS

Para efetuar os testes de funcionalidades e validação das operações propostas no protótipo AFIS *peer-to-peer* foram construídos módulos de software para a execução das seguintes funções:

- Localização de prontuários biométricos por meio do número identificador, CPF;
- Transmissão do *template* biométrico para o nó remoto, proprietário do prontuário;
- Inserção na lista de processamento correspondente, Verificação ou Identificação;
- Cálculo de custo para cada item, considerando as métricas dos nós solicitantes;
- Confronto de Verificação e Identificação.

As funcionalidades listadas são iniciadas como um serviço no sistema operacional Windows 7 nomeado como “Rafis”, sendo controlado por uma aplicação no *tray* do sistema que permite acesso à tela de configuração, tela de envio de *template* e monitoramento das listas de solicitações e confrontos do AFIS (Imagem 5.1). Os dados manipulados pela aplicação são acessados em tabelas do banco de dados “AFIS” no MySQL.

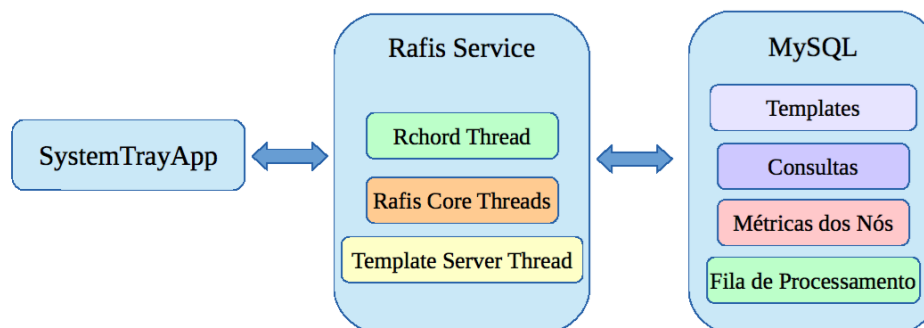


Imagem 5.1 - Esquema simplificado dos módulos e tabelas criadas para o funcionamento do AFIS *peer-to-peer*.

O serviço “Rafis” possui três módulos, carregados como *threads* que executam as seguintes funções:

- ***Rchord Thread***: executa a carga dos identificadores dos prontuários disponíveis na base de dados AFIS do nó, associando o CPF listado ao host local e permitindo a localização destes dados biométricos da rede *Chord*. Esta *thread* monitora as solicitações de pesquisa oriundas deste nó (Tabela de Consultas), localizando prontuários na rede e enviando *templates* a serem confrontados para o nó destino, incrementando a métrica de disponibilidade (DN) do host remoto. Além destas tarefas, é feita continuamente a verificação da Fila de Processamento

local, retornando os resultados processados pelo módulo AFIS (*Rafis Core Thread*) para os nós solicitantes.

- ***Rafis Core Threads:*** são duas *threads* que executam as operações de verificação e identificação do AFIS, carregando e gerenciando os *templates* disponíveis na base local para efetuar os confrontos biométricos. Consultando a Fila de Processamento no banco de dados as *threads* executam continuamente as entradas disponíveis: entradas de confronto de verificação (1:1) na ordem de chegada; entradas de confronto de identificação (1:n) considerando como ordem de processamento as entradas de menor custo. Visando a interoperabilidade com outros AFIS a função deste módulo do software poderá ser desabilitada e implementada por outra ferramenta, vez que os registros a serem processados ficam disponíveis em Banco de Dados com os templates no formato ISO.

- ***Template Server Thread:*** recebe respostas das solicitações locais, *templates* e solicitações de confronto de outros nós. Esta *thread* disponibiliza uma porta de conexão para os *hosts* remotos. Ao receber uma solicitação de confronto, a requisição é persistida na fila correspondente, verificação ou identificação. Ao passo que recebe solicitações de outros nós, a identificação destes *hosts* é armazenada juntamente ao tamanho das bases de dados (NP) informado nas mensagens. Os números de requisições de confronto (NQ) e número de respostas recebidas (NR) são incrementadas a cada mensagem recebida compondo a tabela de Métrica dos Nós.

Os dados biométricos trocados entre os nós são encaminhados via TCP/IP por meio de *sockets*. O módulo *Rchord Thread* exerce o papel de cliente no canal de comunicação entre os nós, o módulo *Template Server Thread* exerce o papel de servidor em cada nó. As mensagens trocadas entre os *peers* possuem uma estrutura única, variando apenas os valores em campos pré-estabelecidos, como apresentado a seguir:

**Operação:** campo que define qual o tipo da mensagem, se é uma consulta de verificação, consulta de identificação ou resposta de uma operação solicitada.

**OpId:** identificador único da mensagem de operação, utilizado para o controle das solicitações recebidas e o envio da resposta ao nó solicitante;

**CPF:** identificador do prontuário a ser consultado, caso operação de verificação. Identificador do *template* compatível na resposta à operação de identificação;

**Node\_DBsize:** apresenta o número de prontuários disponíveis no nó emissor, utilizado quando a mensagem é uma operação de consulta;

**Id\_Dedo:** define a qual dedo o *template* enviado corresponde;

**Template\_ISO:** *Template* armazenado no formato ISO/IEC 19794-2:2011.

**No\_Destino:** armazena o nome do nó de destino, no formato FQDN (Fully Qualified Domain Name);

**No\_Origem:** armazena o nome do nó de origem, no formato FQDN (Fully Qualified Domain Name);

**Resultado:** utilizado nas mensagens de resposta, define o resultado das operações solicitadas especificando o sucesso ou não nas operações de verificação e identificação;

**Imagem:** arquivo de imagem do arquivo biométrico compatível, encaminhado na resposta após o sucesso no confronto entre os *templates*.

**Score:** pontuação alcançada nos confrontos de verificação e identificação para o *template* encaminhado.

Na execução dos testes de operação foram configuradas 5 máquinas virtuais com o sistema Windows 7 (32 *bits*) instalado em cada uma. Nestes *hosts* foram configurados, o aplicativo desenvolvido com a pré-instalação do *Framework* .NET versão 4.5 e o Servidor MySQL, armazenando as tabelas necessárias para operação do sistema, disponibilizado por um script SQL. Cada máquina representava um Estado participante da rede AFIS *peer-to-peer*, sendo estes identificados como representantes dos estados RO, AM, RJ, RR e MT, dentre os demais inseridos aleatoriamente. Os nomes de cada nó seguindo do nome de domínio “*rafis.net*” foi utilizado para identificação dos pontos participantes.

Para validar a aplicação desenvolvida com a proposta inicialmente apresentada foram inseridos 1610 registros de prontuários fictícios distribuídos entre os nós, compostos de números de CPF aleatórios<sup>3</sup>. Os *templates* reais a serem pesquisados ficaram armazenados no *peer* “*RO.afis.net*”, sendo este o nó a ser consultado para efeito dos testes. 349 prontuários biométricos foram utilizados da base de dados do Estado de Rondônia. Cada registro inserido foi considerado único na rede e o tratamento de colisões durante a inserção dos prontuários impedia o caso de prontuários de um mesmo CPF em nós distintos. O tratamento desta exceção

---

3 Os números CPF atribuídos aos prontuários fictícios foram criados aleatoriamente sem considerar o formato dos dígitos verificadores, visando facilitar a implementação de teste.

e o encadeamento dos prontuários neste caso não foi implementada, devido ao limitado tempo e o foco ter sido dado no desenvolvimento das funcionalidades principais da proposta, deixando este recurso para versões futuras.

## 5.1. LOCALIZAÇÃO DE TEMPLATES

O cliente do protótipo é apresentado em uma *Windows Form* acessível por meio do aplicativo na bandeja do desktop (*SystemTrayApp*). Através dos campos na parte inferior da tela é possível selecionar o arquivo biométrico em formato de imagem a ser processado (extensão .jpg ou .obj), escolher o tipo de operação desejada, informar qual o CPF a ser confrontado no caso de verificação e qual o Estado a ser consultado no caso de Identificação. Permite-se ainda definir a qual dedo a imagem selecionada pertence. Nesta tela também é possível visualizar a lista de consultas e a fila de processamento local (Imagem 5.2).

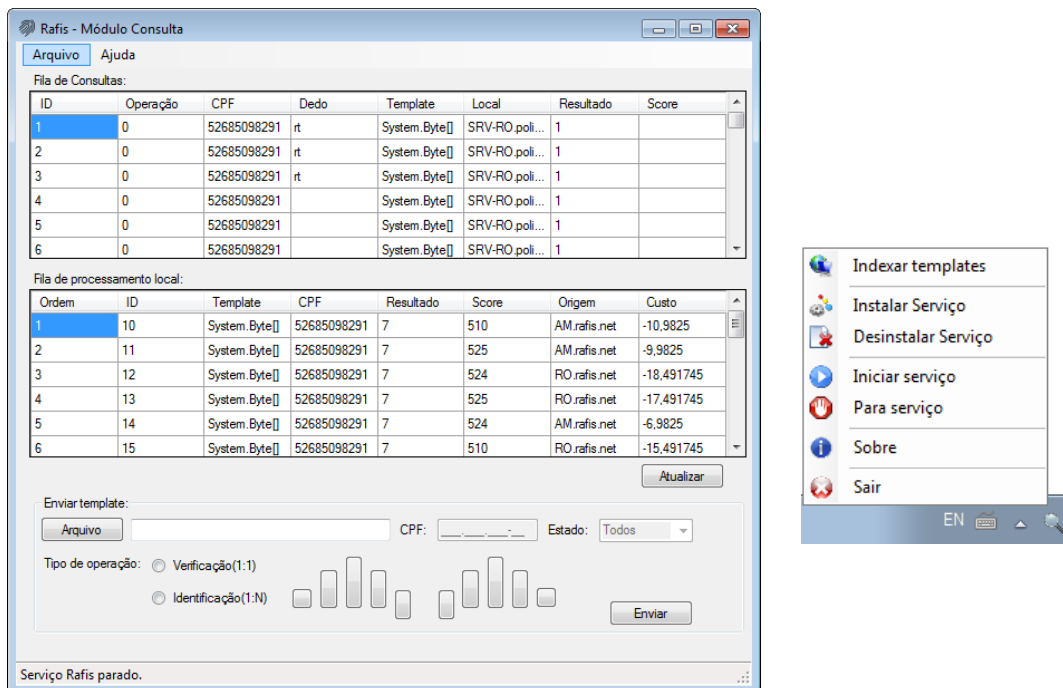


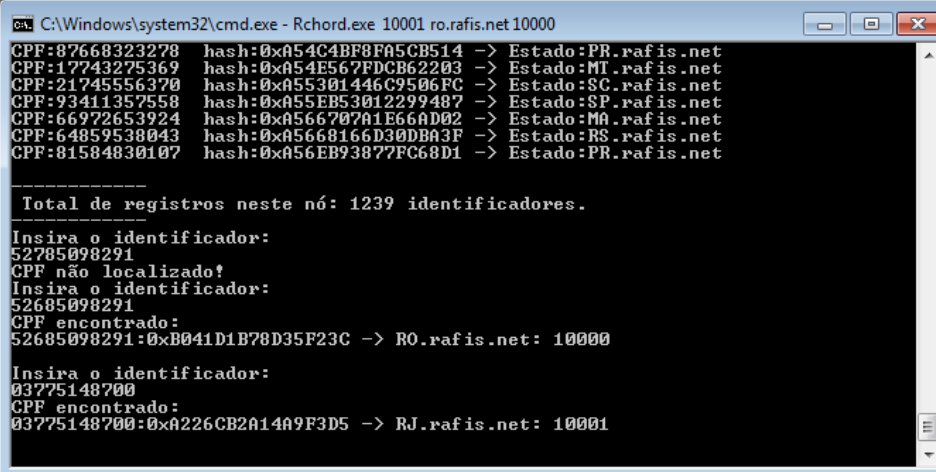
Imagem 5.2 - Interface desenvolvida para controlar o processo “Rafis” e tela de monitoramento das consultas, fila de processamento e cliente para consulta de *templates*.

Caso o tipo de operação selecionado seja a verificação, o campo de preenchimento “CPF” é habilitado para inserção do CPF a ser pesquisado. Uma vez inserido os dados informados, clicando-se em “Enviar” os dados são inseridos em uma tabela de consultas no banco MySQL.

A partir da inicialização do serviço “Rafis”, o módulo “*Rchord Thread*” monitora as novas entradas na tabela de consultas. Para cada nova entrada é executada a busca pelo CPF informado junto à rede *Chord*, recebendo como retorno o nome de *host* no formato FQDN.

Uma versão em linha de comando (Imagem 5.3) foi programada para executar funções de teste e simulações como nós remotos no ambiente virtualizado, tal ferramenta permite a pesquisa, visualização, inserção de prontuários e nós virtuais e interação com a aplicação principal, facilitando os testes do protótipo.

O módulo de pesquisa e localização dos prontuários na rede AFIS *p2p* foi construído sobre a biblioteca *Nchord*, resultando em uma aplicação de linha de comando “*Rchord.exe*”, utilizada para acessar o índice *Chord*, possibilitando a pesquisa manual e adição de prontuários para os testes de operação aqui descritos.



```
C:\Windows\system32\cmd.exe - Rchord.exe 10001 ro.rafis.net 10000
CPF:87668323278 hash:0xA54C4BF8F05CB514 -> Estado:PR.rafis.net
CPF:17743275369 hash:0xA54E567FDCB62203 -> Estado:MI.rafis.net
CPF:21745556370 hash:0xA55301446C9506FC -> Estado:SC.rafis.net
CPF:93411357558 hash:0xA55EB53012299487 -> Estado:SP.rafis.net
CPF:66972653924 hash:0xA566707A1E66AD02 -> Estado:MA.rafis.net
CPF:64859538043 hash:0xA5668166D30DBA3F -> Estado:RS.rafis.net
CPF:81584830107 hash:0xA56EB93877FC68D1 -> Estado:PR.rafis.net

-----
Total de registros neste nó: 1239 identificadores.
-----
Insira o identificador:
52785098291
CPF não localizado!
Insira o identificador:
52685098291
CPF encontrado:
52685098291:0xB041D1B78D35F23C -> R0.rafis.net: 10000

Insira o identificador:
03775148700
CPF encontrado:
03775148700:0xA226CB2A14A9F3D5 -> RJ.rafis.net: 10001
```

Imagem 5.3 - Versão do módulo “*Rchord*” em linha de comando, utilizado para inserção de templates e nós virtuais, apresentando a pesquisa e localização de prontuários pelo mapeamento CPF, FQDN.

Localizando o prontuário atribuído ao CPF pesquisado, o módulo “*Rchord Thread*” formata a mensagem de consulta com os dados fornecidos, estabelece um canal de comunicação entre os nós, utilizando uma porta específica (porta 7777 por padrão), e envia estes dados para o *host* remoto, entregando ao módulo “*Template Server Thread*” no destino.

Durante os testes ficou constado a funcionalidade de pesquisa da solução proposta, retornando o FQDN do nó proprietário do CPF consultado. O correto funcionamento deste módulo da ferramenta é o que viabiliza as operações de confronto de verificação, permitindo localizar um prontuário específico na rede *peer-to-peer*, da forma como proposto na concepção do sistema.

## 5.2. PROPAGAÇÃO DE TEMPLATES E EXECUÇÃO DE CONFRONTOS

Como parte do serviço “Rafis”, a *thread* do servidor de *templates* fica ativa continuamente recebendo as mensagens endereçadas ao nó local. Ao executar uma consulta de verificação, utilizando a parte inferior da tela principal da ferramenta (Imagem 5.4), pudemos testar a funcionalidade do módulo cliente do AFIS *peer-to-peer* proposto. Durante os testes, imagens de impressões digitais no formato .JPG e .OBJ foram submetidas, sendo selecionada a opção “Verificação (1:1)” e o botão referente ao dedo da digital.

Em cada teste, após o preenchimento dos dados de CPF, ao clicar no botão enviar a aplicação processava a imagem selecionada gerando o *template* ISO deste arquivo. Era então incrementada a métrica DN (Disponibilidade do Nó) do nó destino, e o conjunto dos *templates* e demais dados do formulário eram armazenados na tabela de consultas local, recebendo *OpID* único.

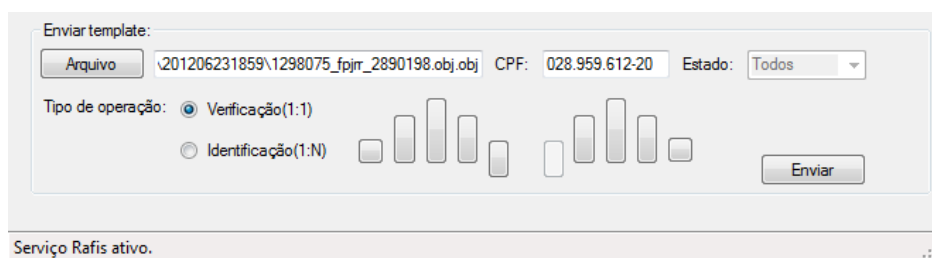


Imagem 5.4 - Seção correspondente ao cliente da aplicação AFIS *peer-to-peer*, na tela principal da ferramenta.

Processando cada nova entrada na tabela de consultas, o módulo “*Rchord Thread*” executava os passos detalhados na seção 4.1. No recebimento de uma nova solicitação, o módulo “*Template Server Thread*” extraía os dados da mensagem, verificando o tipo de operação no campo correspondente. Caso a operação fosse de verificação, os dados recebidos eram inseridos na fila de processamento de verificação e processados pelo módulo “*Rafis Core*”, na *thread* de verificação. Caso a operação informada fosse de Identificação, os dados recebidos eram inseridos na tabela de identificação e o custo desta entrada processado, atualizando este valor no registro correspondente do banco. Tal custo, como detalhado anteriormente, foi calculado considerando a ordem de inserção na fila e os valores presentes na tabela de Métrica dos Nós. Caso a mensagem recebida fosse uma resposta a uma solicitação anterior oriunda deste nó, a entrada correspondente ao valor *OpID* na tabela de consultas era atualizada com os dados recebidos.

Os testes executados foram feitos entre os nós “AM” e “RO”, monitorados através dos registros atualizados no banco de dados e pelos *logs* das *threads* em operação. Três arquivos de

log ficam disponíveis na pasta raiz da aplicação, um arquivo para cada *thread* do serviço “Rafis”, são eles:

- Rafis.log: registro de operação do módulo “*Rchord Thread*”;
- RafisCore.log: registro de operação do módulo “*Rafis Core Thread*”;
- TemplateServer.log: registro de operação do módulo “*Template Server Thread*”.

Consultando os registros nos arquivos, “Rafis.log” do nó de origem e “TemplateServer.log” do no destino, foi possível atestar respectivamente o sucesso no envio e no recebimento dos *templates* biométricos oriundos da consulta verificação.

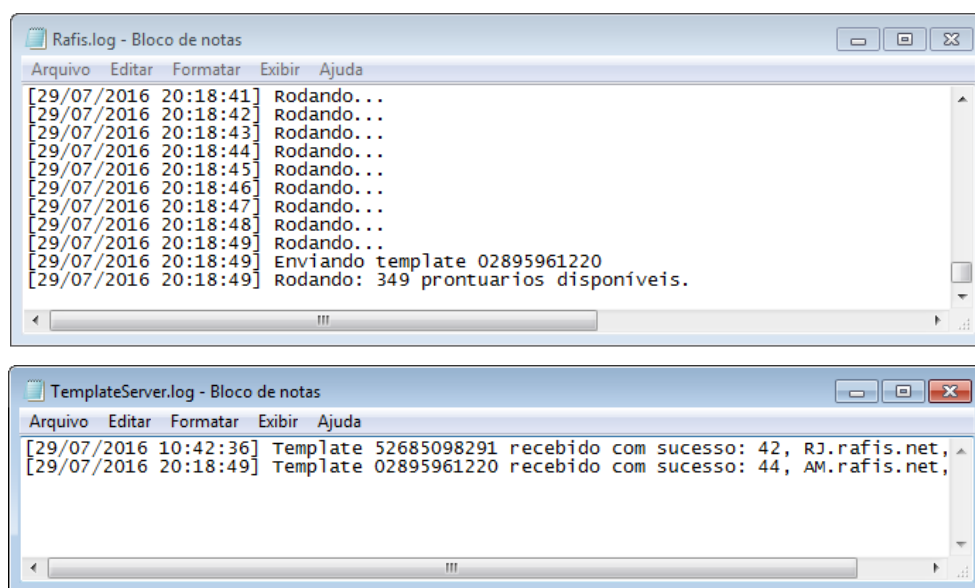


Imagem 5.5 - Conteúdo constatado nos arquivos de *log*, apresentando o envio e o recebimento do *template* submetido a consulta.

Os registros constatados demonstraram o encaminhamento correto da consulta entre os nós origem (*AM.rafis.net*) e destino (*RO.rafis.net*). Após o recebimento com sucesso da mensagem, persistido tais dados na tabela de confronto de verificação, o módulo “*Rafis Core Thread*” executou o método *verify()*. O algoritmo AFIS e processou a entrada, confrontando o *template* encaminhado com o *template* local associado ao CPF informado. O resultado desta operação foi persistido, atualizando a entrada processada, e na sequência gravado no arquivo de *log* “RafisCore.log” (Imagem 5.6).

Após o resultado do teste ser registrado na fila de verificação, o resultado foi atualizado e a operação estava disponível para ser encaminhado de volta ao nó solicitante pelo módulo “*Rchord Thread*”. A cada ciclo de operação este módulo cliente monitorou as entradas já

processadas e verificou os respectivos resultados, respondendo ao nó remoto origem de cada solicitação.

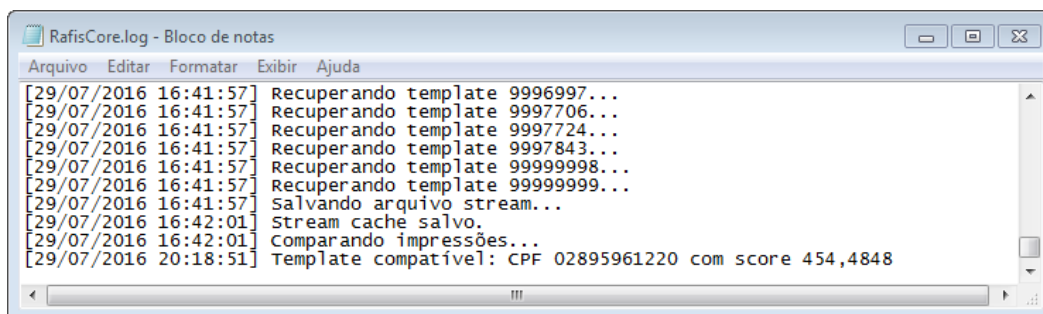


Imagem 5.6 - Conteúdo constatado no arquivo “RafisCore.log”, apresentando o resultado do confronto de Verificação.

A resposta encaminhada era recebida de volta pelo nó de origem pelo módulo “*Template Server Thread*”, que processava os dados desta resposta atualizando a entrada na tabela de consultas, encerrando assim a operação de verificação.

| ID | Operação | CPF         | Dedo | Template      | Local           | Resultado | Score |
|----|----------|-------------|------|---------------|-----------------|-----------|-------|
| 40 | 0        | 52685098291 |      | System.Byte[] | RO.rafis.net... | 1         | 21    |
| 41 | 0        | 52685098291 | rt   | System.Byte[] | RO.rafis.net... | 1         | 38    |
| 42 | 0        | 52685098291 | rt   | System.Byte[] | RO.rafis.net... | 1         | 38    |
| 43 | 0        |             | rt   | System.Byte[] |                 | 0         | 0     |
| 44 | 0        | 02895961220 | rt   | System.Byte[] | RO.rafis.net... | 1         | 454   |

Imagem 5.7 - Tela de monitoramento da tabela de consultas do nó “AM”, constando o resultado de solicitações para o nó “RO”.

As operações descritas nesta seção atestam a funcionalidade da ferramenta em transmitir as solicitações de confronto entre nós, utilizando o formato de mensagem aqui proposto. Ao receber as mensagens, os módulos do serviço “Rafis” executaram as funções de registro e verificação da forma como esperada, processando e respondendo aos nós solicitantes os resultados respectivos.

As operações relativas aos confrontos de identificação sofreram tratamento semelhante ao detalhado nesta seção. A diferença neste caso está no cálculo do custo para cada nova entrada na fila de identificação, que considera o custo de cada entrada para a organização da fila de processamento. Os resultados dos testes desta operação são apresentados a seguir.

### 5.3. ESCALONAMENTO DA FILA DE IDENTIFICAÇÃO

Para executar uma consulta do tipo Identificação (1:n) no aplicativo desenvolvido, foram efetuadas consultas a partir dos nós AM, RR, RJ e MT, destinado ao nó RO. Estes nós, como descrito anteriormente, foram configurados em um ambiente com máquinas virtuais, e



efetuaram consultas de prontuários na rede e solicitações de confronto para serem processadas e monitoradas no *host* “RO.rafis.net”.

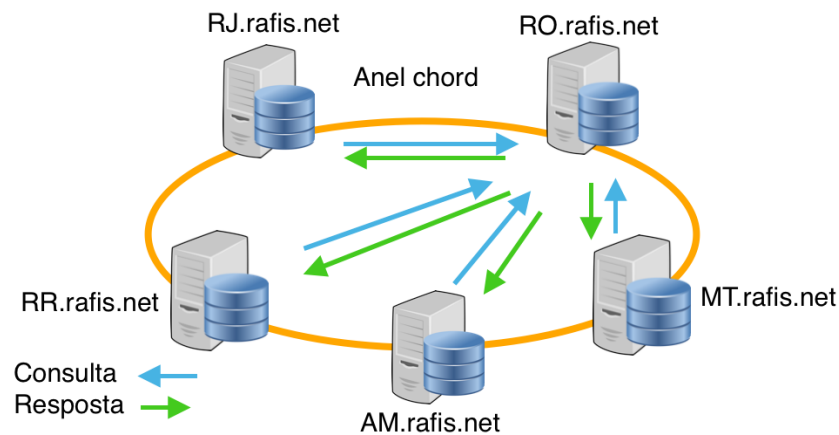


Imagem 5.8 - Esquema apresentando a rede *peer-to-peer*, com os nós participantes representando Estados.

Durante a operação *p2p*, efetuando uma consulta a partir de um nó específico, como exemplo no *host* “AM.rafis.net”, foi utilizado a tela de consulta da aplicação principal. Da mesma maneira que a operação de verificação, o usuário do sistema deve selecionar um arquivo de imagem digital a ser pesquisada. Ao marcar a opção “Identificação (1:n)” o componente *ComboBox* é habilitado apresentando os nomes dos *hosts* conhecidos, aqueles presentes na tabela de Métricas dos Nós, ou nós pré-cadastrados por acordo de cooperação. Nesta operação de busca não se sabe o ID da digital submetida, neste caso o campo CPF é desabilitado.

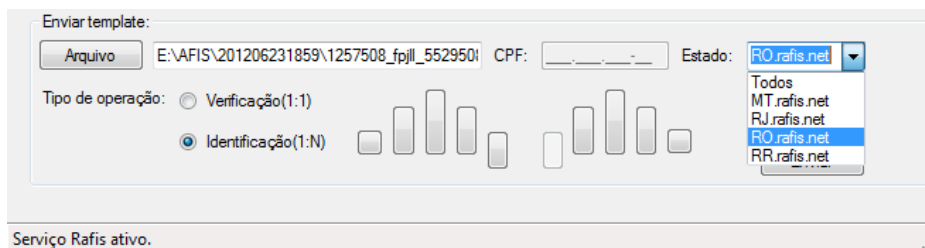


Imagem 5.9 - Tela de consulta no *host* “AM.rafis.net” configurada para enviar consulta ao *host* “RO.rafis.net”.

Ao definir o destino da consulta, clicando-se em “Enviar” a aplicação insere os dados na tabela de consultas do nó local, alterando a métrica DN (Disponibilidade do Nó) do nó destino. A partir deste ponto a *thread* “Rchord” não executa a pesquisa de CPF na rede *Chord*, mas sim efetua a transmissão dos dados diretamente para o nó destino, neste caso “RO.rafis.net”.

Um detalhe importante a ser salientado é que, caso a opção “Todos” da *ComboBox* fosse selecionada, a aplicação cliente replicaria a solicitação para cada *host* na fila de consultas,

encaminhando solicitações para todos os nós presentes na listagem. Esta configuração permite efetuar a pesquisa de uma digital em todos os nós conhecidos da rede. O custo da pesquisa e de processamento neste caso seria dividida entre os nós da rede, de forma distribuída a rede *p2p* procederia o confronto do *template* encaminhado em cada um dos nós contatados, distribuindo também a carga da identificação em todos os nós. É indubitável que esta abordagem é mais eficiente do que se a pesquisa fosse executada em um único nó.

Seguindo com os testes foi possível confirmar o envio e o recebimento das mensagens de comunicação, da mesma forma que descrito na seção 4.2. A diferença estava na forma que a “*Template Server Thread*” tratava as mensagens recebidas, neste caso o código servidor extraía a mensagem, inserindo seu conteúdo na fila de identificação, incrementando a métrica dos nós emissores e calculando o custo para cada entrada. Ao todo foram executadas 30 solicitações de confronto de identificação de cada *host* para o nó “RO”.

**Tabela 5.1 - Fila De Identificação Do Host “Ro.Rafis.Net”.**

| Nova Ordem | OC | OpId | Template      | CPF      | Score | Nó Origem    | Custo     |
|------------|----|------|---------------|----------|-------|--------------|-----------|
| 1          | 4  | 1    | System.Byte[] | 526***** | 525   | MT.rafis.net | -19,9319  |
| 2          | 6  | 2    | System.Byte[] | 526***** | 510   | MT.rafis.net | -17,9319  |
| 3          | 2  | 1    | System.Byte[] | 526***** | 525   | RR.rafis.net | -13,44125 |
| 4          | 13 | 3    | System.Byte[] | 526***** | 525   | MT.rafis.net | -10,9319  |
| 5          | 1  | 1    | System.Byte[] | 526***** | 510   | AM.rafis.net | -10,9025  |
| 6          | 8  | 1    | System.Byte[] | 526***** | 510   | RJ.rafis.net | -9,9324   |
| 7          | 3  | 2    | System.Byte[] | 526***** | 524   | AM.rafis.net | -8,9025   |
| 8          | 7  | 2    | System.Byte[] | 526***** | 525   | RR.rafis.net | -8,44125  |
| 9          | 16 | 4    | System.Byte[] | 526***** | 510   | MT.rafis.net | -7,9319   |
| 10         | 11 | 2    | System.Byte[] | 526***** | 510   | RJ.rafis.net | -6,9324   |
| 11         | 5  | 3    | System.Byte[] | 526***** | 524   | AM.rafis.net | -6,9025   |
| 12         | 9  | 3    | System.Byte[] | 526***** | 510   | RR.rafis.net | -6,44125  |
| 13         | 12 | 3    | System.Byte[] | 526***** | 525   | RJ.rafis.net | -5,9324   |
| 14         | 20 | 5    | System.Byte[] | 526***** | 525   | MT.rafis.net | -3,9319   |
| 15         | 15 | 4    | System.Byte[] | 526***** | 524   | RJ.rafis.net | -2,9324   |
| 16         | 10 | 4    | System.Byte[] | 526***** | 525   | AM.rafis.net | -1,9025   |
| 17         | 17 | 5    | System.Byte[] | 526***** | 525   | RJ.rafis.net | -0,9324   |
| 18         | 23 | 6    | System.Byte[] | 526***** | 510   | MT.rafis.net | -0,9319   |
| 19         | 14 | 5    | System.Byte[] | 526***** | 525   | AM.rafis.net | 2,0975    |
| 20         | 27 | 7    | System.Byte[] | 526***** | 38    | MT.rafis.net | 3,0681    |
| 21         | 21 | 4    | System.Byte[] | 526***** | 510   | RR.rafis.net | 5,55875   |
| 22         | 18 | 6    | System.Byte[] | 526***** | 525   | AM.rafis.net | 6,0975    |
| 23         | 22 | 5    | System.Byte[] | 526***** | 524   | RR.rafis.net | 6,55875   |
| 24         | 25 | 6    | System.Byte[] | 526***** | 21    | RJ.rafis.net | 7,0676    |
| 25         | 19 | 7    | System.Byte[] | 526***** | 510   | AM.rafis.net | 7,0975    |
| 26         | 28 | 7    | System.Byte[] | 526***** | 38    | RJ.rafis.net | 10,0676   |
| 27         | 26 | 6    | System.Byte[] | 526***** | 38    | RR.rafis.net | 10,55875  |
| 28         | 24 | 8    | System.Byte[] | 526***** | 38    | AM.rafis.net | 12,0975   |

|    |    |    |               |          |     |              |         |
|----|----|----|---------------|----------|-----|--------------|---------|
| 29 | 29 | 9  | System.Byte[] | 289***** | 454 | AM.rafis.net | 17,0975 |
| 30 | 30 | 10 | System.Byte[] | 526***** | 36  | AM.rafis.net | 18,0975 |

A tabela de métrica dos nós durante a operação foi sendo atualizada a cada solicitação recebida, servindo como as variáveis para o cálculo dos custos das entradas. Esta tabela armazenava o identificador dos nós solicitantes, as variáveis DN, NR, NQ e NP da função de Custo.

| NodeId | Name         | DispNode | ReqNode | RespNode | NumPront |
|--------|--------------|----------|---------|----------|----------|
| 1      | AM.rafis.net | 12       | 10      | 12       | 500      |
| 3      | RR.rafis.net | 19       | 6       | 12       | 250      |
| 4      | RJ.rafis.net | 18       | 7       | 18       | 480      |
| 5      | MT.rafis.net | 25       | 7       | 23       | 380      |
| NULL   | NULL         | NULL     | NULL    | NULL     | NULL     |

Imagem 5.10 - Tabela de métricas dos nós com os respectivos valores das variáveis durante a operação.

O processamento de identificação executado pela *thread* “Rafis Core” de identificação processava as entradas listando os registros na ordem de menor custo. Ao passo que os registros eram processados, o estado de cada entrada era atualizado com o resultado da operação, juntamente à identificação do *template* compatível e o *score* atingido no confronto (coluna *score* da Tabela3).

Finalizando o processamento os registros na fila de identificação, os dados de cada entrada, relativos ao *template* compatível e o *score* resultante do confronto, foram tratados pela *thread* cliente “Rchord” compondo uma nova mensagem de resposta para o nó solicitante, encerrando a operação.

As operações efetuadas permitiram constatar que a fórmula proposta para escalonamento da fila de identificação apresentou um reordenamento mais justo para os nós que disponibilizaram recursos. Os nós que participaram mais na rede, exemplo o nó “MT” possuíam melhores métricas, ganhando vantagens no processamento de suas requisições. Durante as simulações percebeu-se a baixa influência do tamanho da base de dados no cálculo do custo nas entradas da fila de identificação. Este efeito foi decorrente do pequeno tamanho das bases utilizadas nos testes frente ao coeficiente utilizado na função junto à variável NP, valor  $5 \cdot 10^{-6}$ .

## 6. CONCLUSÕES

O trabalho aqui descrito apresentou a concepção de um sistema de localização, compartilhamento e confronto de *templates* biométricos em uma rede *peer-to-peer*, utilizando o protocolo Chord para distribuição dos identificadores. Também apresentou uma proposta de escalonamento para filas de confronto baseada em custos, com métricas e pesos calculados para cada entrada no processo de identificação do AFIS. Esta proposta calcula um custo para cada solicitação de operação entre nós, considerando parâmetros como ordem de chegada, disponibilidade, número de requisições, número de respostas e tamanho da base de *templates*.

Tal solução é apresentada como uma alternativa para confronto entre *templates* de nós distintos, podendo ser utilizado em consultas de verificação (1:1) ou pesquisas de identificação (1:n) entre *peers*. A aplicação desenvolvida como prova de conceito viabilizou a pesquisa de prontuários biométricos em rede, permitindo o confronto de dados biométricos entre sistemas AFIS distintos, sem uma estrutura centralizada ou controlada de forma global. Este conceito permite um modelo não hierárquico de comunicação entre sistemas do tipo, atendendo as particularidades políticas interinstitucionais no Brasil. Esta aplicação poderia integrar Estados, permitindo consultas entre bases biométricas e possibilitando a identificação de indivíduos registrados em outras unidades da federação.

Uma ferramenta AFIS funcional baseada no projeto SourceAFIS foi concebida, tal artefato serviu de base para a construção do módulo AFIS para os testes em rede. Este módulo de software foi programado para gerar *templates* e processar um pequeno conjunto de impressões digitais disponibilizados pelo Governo do Estado de Rondônia para estudos neste trabalho. As amostras utilizadas foram a base utilizada para execução dos testes de verificação e identificação no módulo AFIS, tanto para o software inicial quanto para a aplicação *peer-to-peer*.

O protocolo *Chord* se mostrou eficiente para localização dos *templates*, mantendo um índice de pesquisa com base nos CPFs dos prontuários existentes em cada nó. Através desta rede *peer-to-peer* foi possível divulgar os prontuários disponíveis em cada *peer* de maneira descentralizada, viabilizando a operação da infraestrutura, ainda que em caso de falhas nos nós. O *Chord* agrega em flexibilidade e eficiência para a ferramenta desenvolvida, dispondo um índice distribuído para mapeamento dos recursos locais para a rede.

O software desenvolvido possibilitou a transferência dos dados biométricos entre os *hosts* da rede, integrando os nós e viabilizando a operação entre AFIS remotos. Suas bases de

dados biométricos acessíveis desta forma atenderam as características levantadas inicialmente, de heterogeneidade e características político-administrativas entre participantes, permitindo o compartilhamento sob demanda dos prontuários cadastrados em cada AFIS local.

Foi possível constatar através de simulações entre os *peers*, compondo filas de identificação aleatoriamente, que a função proposta permite um reordenamento dos itens recebidos de maneira satisfatória, escalonando o processamento dos *templates* no sistema AFIS dos nós participantes. Tal abordagem valoriza os nós que disponibilizam mais os seus recursos na rede, tornando o compartilhamento entre nós uma operação sustentável na oferta de recursos biométricos na rede, pois os nós que mais disponibilizam consultas e dados biométricos recebem melhores custos nas solicitações em nós remotos. Atendendo desta maneira o princípio de sustentabilidade e balanceamento dos recursos e demandas do ambiente em redes do tipo.

Fica proposto para trabalhos futuros a avaliação e o impacto da carga dos dados biométricos junto ao índice de pesquisa do Chord, permitindo o compartilhamento de *templates* no formato ISO/IEC 19794-2:2011 diretamente no *peer* da rede. Poderia ser avaliado também o aspecto de segurança desta abordagem, que em tese garantiria a privacidade dos prontuários compartilhados.

## **6.1. TRABALHOS PUBLICADOS**

A pesquisa desenvolvida nesta dissertação resultou além do software prova de conceito em dois artigos publicados em eventos acadêmicos da Sociedade Brasileira de Computação. Tal lastro de pesquisa e de estudos serviu como base para a escrita e aperfeiçoamento do trabalho aqui apresentado, o conjunto produzido no período de pesquisa é apresentado a seguir:

Santos, C. G. C.; De Sousa Júnior, R. T.; De Deus, F. E. G. Prototipação E Validação De Um Afis De Código Aberto Utilizando A Base De Identificação Criminal Do Estado De Rondônia. Anais Do Wgid - V Workshop De Gestão De Identidades Digitais, 2015, XV Simpósio Brasileiro Em Segurança Da Informação E De Sistemas Computacionais Sbseg 2015, V. 1. Pp. 575-586.

Santos, C. G. C.; De Sousa Júnior, R. T. Estrutura e Funcionamento de um Sistema de Identificação. Biométrico Peer-to-peer baseado no Protocolo Chord. Anais Do Wgid - VI Workshop De Gestão De Identidades Digitais, 2016, XVI Simpósio Brasileiro Em Segurança Da Informação E De Sistemas Computacionais Sbseg 2016, V. 1. Pp. 660-671.

## REFERÊNCIAS BIBLIOGRÁFICAS

- Abdul-Rahman, Alfarez; Hailes, Stephen. Supporting Trust In Virtual Communities. In: System Sciences, 2000. Proceedings Of The 33rd Annual Hawaii International Conference On. Ieee, 2000. P. 9 Pp. Vol. 1.
- Androutsellis-Theotokis, Stephanos; Spinellis, Diomidis. A Survey Of Peer-To-Peer Content Distribution Technologies. *Acm Computing Surveys (Csur)*, V. 36, N. 4, P. 335-371, 2004.
- Adderley, Richard; Bond, John W. The Effects Of Deprivation On The Time Spent Examining Crime Scenes And The Recovery Of Dna And Fingerprints. *Journal Of Forensic Sciences*, V. 53, N. 1, P. 178-182, 2008.
- Ayyasamy, S.; Sivanandam, S. N. A Cluster Based Replication Architecture For Load Balancing In Peer-To-Peer Content Distribution. *Arxiv Preprint Arxiv:1009.4563*, 2010.
- Brasil. Ministério Da Justiça (2016). Página Ric – Registro De Identidade Civil. Disponível Em: <[Http://Justica.Gov.Br/Acesso/Governanca/Ric](http://Justica.Gov.Br/Acesso/Governanca/Ric)>. Acesso Em: 01 De Julho De 2016.
- Chawachat, Jakarin; Fakcharoenphol, Jittat. A Simpler Load-Balancing Algorithm For Range-Partitioned Data In Peer-To-Peer Systems. *Networks*, V. 66, N. 3, P. 235-249, 2015.
- Cencini, Andrew. (2009) Nchord 1.0.1.0, Technical Documentation And Tutorial.
- Cjis, Criminal Justice Information Services. Eletronic Biometric Transmission (Ebts). Julho De 2013. Federal Bureau Of Investigation, Criiminal Justice Information Services Division. Claksburg, Wv.
- Chapman, Will Et Al. Latent Interoperability Transmission Specification. *Nist Special Publication*, V. 1152, 2013.
- De Macedo Rodrigues, Ramysés; Ribeiro, Rafael Oliveira. Estudo De Performance De Algoritmos De Verificação De Impressões Digitais Aplicáveis A Smart-Card. In: *Anais Da Conferência Internacional De Ciências Forenses Em Multimídia*. P. 49.
- De Macedo Rodrigues, Ramyses; Costa, Marly Guimarães Fernandes; Costa Filho, Cicero Ferreira Fernandes. Fingerprint Verification Using Characteristic Vectors Based On Planar Graphics. *Signal, Image And Video Processing*, V. 9, N. 5, P. 1121-1135, 2015.
- De Macedo Rodrigues, Ramyses; Costa, Marly Guimarães Fernandes; Costa Filho, Cicero Ferreira Fernandes. Verificação De Impressões Digitais Usando Modelo De Vetor Característico Baseado Em Grafos Planares. In: *Anais Do X Simpósio Brasileiro De Automação Inteligente*, Pp. 989–994. Universidade Federal De São João Del-Rei, São João Del-Rei, Mg, Brasil, Setembro 2011 (2011)
- Dorizzi, Bernadette Et Al. Fingerprint And On-Line Signature Verification Competitions At Icb 2009. In: *International Conference On Biometrics*. Springer Berlin Heidelberg, 2009. P. 725-732.
- Epsztejn, Gabriel; Duarte, Otto Carlos M.B. Proposta De Rede P2p Organizada Por Índices, Grupo De Teleinformática E Automação, Universidade Federal Do Rio De Janeiro, Ufrj, Rio De Janeiro, Brasil, 2006.
- Galar, Mikel Et Al. A Survey Of Fingerprint Classification Part I: Taxonomies On Feature Extraction Methods And Learning Models. *Knowledge-Based Systems*, V. 81, P. 76-97, 2015.

- Ganesan, Prasanna; Bawa, Mayank; Garcia-Molina, Hector. Online Balancing Of Range-Partitioned Data With Applications To Peer-To-Peer Systems. In: Proceedings Of The Thirtieth International Conference On Very Large Data Bases-Volume 30. Vldb Endowment, 2004. P. 444-455.
- Garofalakis, John; Michail, Theofanis-Aristofanis. Load Balancing In A Cluster-Based P2p System. In: Informatics, 2009. Bci'09. Fourth Balkan Conference In. Ieee, 2009. P. 133-138.
- Godfrey, Brighten Et Al. Load Balancing In Dynamic Structured P2p Systems. In: Infocom 2004. Twenty-Third Annual Joint Conference Of The Ieee Computer And Communications Societies. Ieee, 2004. P. 2253-2262.
- Gopalakrishnan, Vijay Et Al. Adaptive Replication In Peer-To-Peer Systems. In: Distributed Computing Systems, 2004. Proceedings. 24th International Conference On. Ieee, 2004. P. 360-369.
- Hendrikx, Ferry; Bubendorfer, Kris; Chard, Ryan. Reputation Systems: A Survey And Taxonomy. Journal Of Parallel And Distributed Computing, V. 75, P. 184-197, 2015.
- Hsiao, Hung-Chang Et Al. Load Balance With Imperfect Information In Structured Peer-To-Peer Systems. Ieee Transactions On Parallel And Distributed Systems, V. 22, N. 4, P. 634-649, 2011.
- Ibge. Censo Demográfico 2010 - Resultados Do Universo. (2010) Disponível Em: [Http://Www.Censo2010.Ibge.Gov.Br/Sinopse/Index.Php?Dados=4&Uf=00](http://www.censo2010.ibge.gov.br/sinopse/index.php?dados=4&uf=00), Acesso Em: 02/06/2016.
- Irum, Sarah Et Al. How To Build An Automated Fingerprint Identification System. In: Biometrics And Security Technologies (Isbast), 2013 International Symposium On. Ieee, 2013. P. 40-47.
- Kamvar, Sepandar D.; Schlosser, Mario T.; Garcia-Molina, Hector. The Eigentrust Algorithm For Reputation Management In P2p Networks. In: Proceedings Of The 12th International Conference On World Wide Web. Acm, 2003. P. 640-651.
- Karger, David R.; Ruhl, Matthias. Simple Efficient Load Balancing Algorithms For Peer-To-Peer Systems. In: Proceedings Of The Sixteenth Annual Acm Symposium On Parallelism In Algorithms And Architectures. Acm, 2004. P. 36-43.
- Komarinski, Peter. Automated Fingerprint Identification Systems (Afis). Academic Press, 2005.
- Koutrouli, Eleni; Tsalgatidou, Aphrodite. Reputation-Based Trust Systems For P2p Applications: Design Issues And Comparison Framework. In: International Conference On Trust, Privacy And Security In Digital Business. Springer Berlin Heidelberg, 2006. P. 152-161.
- Lua, Eng Keong Et Al. A Survey And Comparison Of Peer-To-Peer Overlay Network Schemes. Ieee Communications Surveys & Tutorials, V. 7, N. 2, P. 72-93, 2005.
- Ludwig, Artulino. (1996). A Perícia Em Local De Crime. Canoas: Ed. Da Ulbra, 1996.
- Maltoni, Davide Et Al. Handbook Of Fingerprint Recognition. Springer Science & Business Media, 2009.
- Mayo, Kristi. Afis Interoperability: Leaders In The Field Of Latent-Print Identification Are Starting To Look For An "Enter Once, Search Many" Solution. Evidence Technology Magazine, V. 6, N. 1, P. 12-16, 2008.
- Midha, Jagriti; Sehgal, Amit. Peer To Peer Network: A Review. Ijrit International Journal Of Research In Information Technology, Volume 2, Issue 5, May 2014, Pg: 658-663

- Mondal, Anirban; Goda, Kazuo; Kitsuregawa, Masaru. Effective Load-Balancing Of Peer-To-Peer Systems. In: Data Engineering Workshop. 2003.
- Moses, K. R. Et Al. Automated Fingerprint Identification System. Fingerprint Sourcebook, United States Of America, National Institute Of Justice, 2010.
- Moura, André Luiz. Uma Proposta Para A Triangulação De Delaunay 2d E Localização Planar De Pontos Em Ocaml. 2006. Tese De Doutorado.
- Patange, Vishakha; Gatade, D. D. Survey Of Load Balancing Approaches In Peer-To-Peer Network. International Journal Of Soft Computing And Engineering, V. 3, N. 2, P. 422-424, 2013.
- Pinyol, Isaac; Sabater-Mir, Jordi. Computational Trust And Reputation Models For Open Multi-Agent Systems: A Review. Artificial Intelligence Review, V. 40, N. 1, P. 1-25, 2013.
- Santos, Clayton Gc; De Sousa Júnior, Rafael T.; De Deus, Flávio Eg. Prototipação E Validação De Um Afis De Código Aberto Utilizando A Base De Identificação Criminal Do Estado De Rondônia. Anais Do Wgid - V Workshop De Gestão De Identidades Digitais, 2015, Xv Simpósio Brasileiro Em Segurança Da Informação E De Sistemas Computacionais Sbseg 2015, V. 1. Pp. 575-586
- Sesp - Secretaria De Estado Da Segurança Pública De São Paulo (2014). São Paulo Ganha Nova Carteira De Identidade, Governo Do Estado De São Paulo, [Http://Www.Ssp.Sp.Gov.Br/Noticia/Lenoticia.Aspx?Id=33409](http://www.ssp.sp.gov.br/noticia/lenoticia.aspx?id=33409), Acesso Em 02/06/2016.
- Spangler, Todd. "Netflix Bandwidth Usage Climbs To Nearly 37% Of Internet Traffic At Peak Hours", 28 May 2015 Ny Digital Editor Ny Digital Editor [Http://Variety.Com/2015/Digital/News/Netflix-Bandwidth-Usage-Internet-Traffic-1201507187/](http://variety.com/2015/digital/news/netflix-bandwidth-usage-internet-traffic-1201507187/)
- Stoica, Ion Et Al. Chord: A Scalable Peer-To-Peer Lookup Service For Internet Applications. 2001. In: Proceedings Of The 2001 Conference On Applications, Technologies, Architectures, And Protocols For Computer Communications, Acm.
- Suryanarayana, Girish; Taylor, Richard N. A Survey Of Trust Management And Resource Discovery Technologies In Peer-To-Peer Applications. 2004.
- Tada, Carlos Henrique Moniwa. Estimativa De Qualidade De Impressões Digitais Utilizando Sistemas De Inferência Fuzzy. 2012.
- Watson, C. Et Al. The Nbis-Ec Software Is Subject To Us Export Control Laws. Nist, Gaithersburg, Md, Usa, Tech. Rep, V. 1, P. 2, 2007.
- Vazan, Robert. (2012) Sourceafis Tutorial, Wikibooks.Org, [Https://En.Wikibooks.Org/Wiki/Sourceafis/Tutorial](https://en.wikibooks.org/wiki/Sourceafis/Tutorial).
- Vieira, M. (2015) Pontos Característicos, [Http://Www.Papiloscopia.Com.Br/Pontos.Html](http://www.papiloscopia.com.br/pontos.html) Acesso Em 10/05/2015.