Sistema Baseado em Microsserviços para Identificação Automática de Impressões Digitais

João P. B. Andrade*, Renan A. Barbosa*, Gabriel Bezerra‡, Francisco I. S. Lima*, Mauro R. C. da Silva§, Paulo A. L. Rego*, José G. R. Maia†, Fernando A. M. Trinta*

* Departamento de Computação, Universidade Federal do Ceará, † Virtual UFC Institute,
‡ Laboratório de Processamento de Imagens, Sinais e Computação Aplicada (LAPISCO)
Federal Institute of Ceará
§ Instituto de Computação, Universidade Estadual de Campinas
Email: jpandrade@alu.ufc.br, {pauloalr, gilvanm}@ufc.br

***Resumo*—Biometria por meio das impressões digitais é a forma mais aceita e difundida de distinguir pessoas no mundo atual, devido aos eficientes e acessíveis equipamentos disponíveis (e.g., em celulares, caixas eletrônicos e trancas eletrônicas). Com base neste contexto e a necessidade do governo de utilizar os dados coletados dos cidadãos para diferenciá-los e identificá-los para os mais diversos objetivos, este trabalho relata a implementação de um sistema de identificação de digitais utilizando conceitos de computação concorrente, microsserviços e big data. O sistema consegue cadastrar um indivíduo com dez capturas de seus 10 dedos em um tempo médio de 25,8 segundos, contando com validação de qualidade de cada imagem, extração de características e armazenamento em diferentes bases de dados. O sistema consegue chegar próximo a 90% de assertividade na identificação (1:N) em menos de 5s em um universo de 10 mil indivíduos, também alcançando 84% de acurácia na verificação (1:1). Este trabalho encontra-se em evolução e demonstra resultados promissores com margem para melhorias.**

I. Introdução

Segundo Liu e Silverman [1], Biometria (do grego *bios*: vida; *metron*: medida) é o uso de características biológicas em mecanismos de identificação que utilizam métodos para verificação ou reconhecimento baseado em medidas anatômicas, fisiológicas e características comportamentais de um indivíduo. Tais medidas lidam com partes do organismo humano que apresentam características distintas para cada indivíduo e com isso não podem ser copiadas ou violadas (e.g., padrões da íris, da retina ou as impressões digitais).

No caso particular da impressão digital, esta representa a técnica biométrica mais aceita no mundo e sua utilização tem se expandido cada vez mais. Por exemplo, *smartphones* já a utilizam como forma de proteção dos dados do proprietário quando da perda ou roubo do dispositivo e sistemas bancários também utilizam impressões digitais como recurso extra de segurança aos usuários de caixas automáticos. Dentre os fatores que favorecem para tal uso, incluem-se: (*i*) o alto grau de precisão no reconhecimento e na capacidade de distinguir indivíduos, pois as impressões digitais são formadas durante os primeiros meses de gestação e mesmos gêmeos idênticos possuem padrões distintos; (*ii*) o crescimento no mercado de dispositivos de baixo custo utilizados para aquisição das digitais, habilitando um grande número de aplicações; e (*iii*) a facilidade no uso dos dispositivos ergonômicos para aquisição da digital.

O advento da informática na biometria determinou o surgimento dos Sistemas Automáticos de Identificação de Impressões Digitais (AFIS - do inglês, *Automated Fingerprint Identification System*) que usam recursos computacionais para obter, armazenar e analisar dados de impressão digital, aumentando a eficiência e eficácia no processo de verificação e identificação de pessoas. Com isso, estes sistemas se tornaram uma ferramenta otimizadora e catalisadora da identificação civil e criminal, facilitando identificar suspeitos, detentos ou indivíduos dentro do sistema de Segurança Pública.

Neste artigo apresentamos um AFIS desenvolvido no escopo de um projeto em parceria com uma instituição governamental, que se baseia em microsserviços e tecnologias de código aberto. Tal proposta é parte de uma iniciativa dessa instituição de agregar serviços computacionais para enriquecer as atividades de Inteligência na área de segurança dos seus cidadãos. O AFIS desenvolvido substitui produtos proprietários com alto custo de licenciamento e contribui a redução de gastos públicos. A partir de um conjunto de requisitos pré-estabelecidos, uma arquitetura de microsserviços foi projetada e implementada para atender aos requisitos previstos em sistemas proprietários existentes no mercado. Os resultados iniciais de experimentos com 10 mil indivíduos mostraram um desempenho promissor, com a verificação atingindo 84% de acurácia, enquanto a identificação alcançando acurácia próxima da verificação com tempo de resposta de 5 segundos a depender das informações fornecidas durante a consulta.

Este artigo está dividido em seis Seções. Em seguida, a Seção 2 apresenta trabalhos relacionados com o presente. Na Seção 3, são apresentados os requisitos definidos para o AFIS proposto, bem como sua arquitetura. A quarta Seção apresenta detalhes na implementação do AFIS e resultados de experimentos. A Seção 5 conclui o artigo e apresenta perspectivas futuras de evolução do trabalho.

II. Trabalhos Relacionados

A biometria digital não é um assunto recente e vários trabalhos relacionados a inovações nos processos de verificação

ou identificação já foram publicados. Peralta et al. [2] propõem uma solução distribuída para verificação e identificação de impressões digitais para lidar com grandes volumes de imagens em um tempo considerado razoável, a solução é adaptável a qualquer algoritmo de *matching* e visa diminuir o tempo de respostas a consultas de identificação sem perda de precisão. Toda a solução é baseada em paralelização de tarefas e computação de alto desempenho, sem trabalhar outros conceitos de sistemas distribuídos, e foram executados testes com 400.000 imagens, resultando em uma escalabilidade linear em relação ao volume de dados e boa adaptabilidade ao uso de hardware. Não é fornecido detalhes sobre o processo de cadastro das digitais, nem subprocessos como extração de minúcias e controle de qualidade, nem indexação de *templates* para facilitar as buscas durante a identificação.

Zhao et al. [3] descrevem um sistema distribuído de identificação de impressões digitais com balanceamento de carga para lidar com extração, armazenamento e acesso simultâneo a *templates* armazenados. Além disso, um algoritmo de extração foi desenvolvido usando a biblioteca *Hadoop Image Processing Interface* para extrair minúcias de maneira rápida e paralela, além de balancear carga no *cluster* do MongoDB, permitindo acesso rápido a um grande número de *templates*. Finalmente, foi utilizado um algoritmo de *matching* proposto em [4], e o sistema foi testado com o conjunto de dados FVC2006 e mostrou resultados promissores em relação à extração de minúcias e acesso aos *templates*. A solução não detalha explicitamente questões como indexação de digitais nem controle de qualidade de imagens cadastradas.

No trabalho de Lastra et al. [5] os autores propõem a paralelização do algoritmo criado por Jiang e Crookes [6] para fazer uso de GPUs, além disso, eles fizeram uma versão utilizando CPUs utilizando paralelismo. O modelo, utilizando uma única GPU, conseguiu ser 15 vezes mais rápido se comparado a um utilizando CPU com *multi-threading*. Ao fazer o uso de 4 GPUs em paralelo, esse número chega a ser 54 vezes superior comparado a utilizar apenas CPU. Para executar os experimentos, foi utilizado a base DB14 com 54.000 digitais [7], e foi criado um *dataset* sintético utilizando o software SFinGe [8] para gerar 800.000 digitais. O trabalho de Lastra et al. [5] se diferencia do atual ao utilizar do *matching* Jiang, enquanto o presente trabalho faz uso do *SourceAFIS*. Além disso, aqui faz-se uso de uma arquitetura baseada em microsserviços.

Cappelli et al. [9] propõem um algoritmo paralelo para identificação de digitais para ser executado em GPUs. Além disso, são criados dois algoritmos para fins de comparação, um sequencial e outro paralelo, ambos para serem executados em CPUs. O algoritmo que é executado em GPUs conseguiu ser 1946 vezes mais rápido que o sequencial, e 207 mais rápido em relação ao paralelo. Para os testes com GPU, foram utilizadas 4 GPUs Tesla C2075 em um computador com um processador Intel XEON. O trabalho de Cappelli et al. [9] se diferencia do presente ao fazer uso de GPUs enquanto o atual trabalho faz apenas de CPUs, e se distingue por não utilizar microsserviços.

III. AFIS PROPOSTO

As subseções a seguir descrevem a visão geral da arquitetura do AFIS proposto, seus requisitos e componentes.

A. Requisitos do AFIS

Na concepção da proposta do AFIS, os requisitos funcionais representam as típicas funcionalidades que tais sistemas fornecem: (i) gerenciamento das digitais de indivíduos, (ii) gerenciamento de usuários do AFIS, (iii) verificação e identificação de indivíduos e (iv) fornecimento de informações de desempenho e auditoria do sistema. Vale a pena frisar que dados pessoais de indivíduos (e.g., nome e data de nascimento) não são gerenciados pelo AFIS. Esta função é de responsabilidade de outros sistemas da instituição governamental. Em outras palavras, ao cadastrar o dado de um indivíduo, o AFIS recebe um código de identificação já existente que permite cruzar suas informações pessoais com os dados biométricos fornecidos.

Em relação aos requisitos não-funcionais, destacam-se a escalabilidade, eficiência, interoperabilidade e abertura, tolerância a falhas e segurança. Em relação aos dois primeiros requisitos, o AFIS proposto poderá gerenciar 9 milhões de indivíduos, sem prejuízo tanto na precisão das respostas quanto na eficiência (tempo de resposta) do sistema, exigindo que o mesmo seja capaz de suportar a demanda de centenas de usuários em paralelo. Já em relação à interoperabilidade e abertura, há previsão que diversos sistemas da instituição possam fazer uso do AFIS. Estes sistemas incluem plataformas móveis usadas por agentes públicos em operações em campo, ou outros sistemas que desejem realizar a autenticação de usuários para acesso a serviços da instituição, como hospitais. Com isso, é necessário que o sistema permita tal integração sem maiores empecilhos. O AFIS também englobará uma série de componentes responsáveis pelas diferentes funções de captura de digitais, verificação e identificação de indivíduos. Estes componentes podem sofrer falhas por inúmeras razões, que porém não devem trazer impacto ao funcionamento do sistema como um todo. Em consequência, o AFIS deve oferecer mecanismos de recuperação de tarefas interrompidas ou mesmo replicação de tarefas/componentes para garantir que processos sejam continuados. O AFIS deve também oferecer garantias de segurança no seu acesso, como autenticação e autorização dos usuários de modo a verificar que um usuário é realmente quem ele afirma ser ou definir o que um determinado usuário pode ou não fazer.

B. Arquitetura

A Figura 1 apresenta uma visão geral da arquitetura do AFIS proposto, composta por seus serviços e componentes, que podem ser replicados para lidar com a variação da carga de trabalho aplicada ao sistema.

O AFIS `API` (`Web Services` e `Load Balancer`) são os pontos de entrada de todas as requisições que chegam ao AFIS. Uma API REST expõe um conjunto de serviços via `HTTP` para acesso por diversos tipos de clientes (e.g., navegadores Web, *tablets* e *smartphones*) que queiram usar os serviços. Este serviço registra todas as requisições no

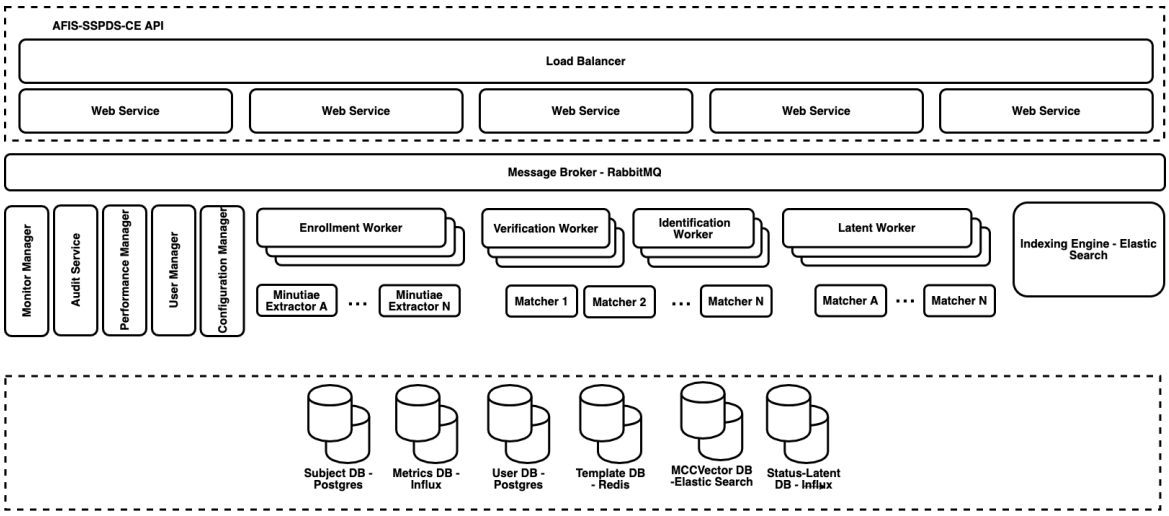

Figura 1. Arquitetura do AFIS proposto

`Message Broker` (`RabbitMQ`, no caso) usando um modelo síncrono de chamada remota de procedimento (do inglês, *Remote Procedure Call - RPC*).

O `Message Broker` é responsável por rotear e enfileirar as requisições dos clientes para os componentes responsáveis por seu correto tratamento. As requisições podem ter diferentes tipos e são encaminhadas para os componentes específicos de tratamento das requisições. As funções de cada componente são descritas a seguir.

O `Monitor Manager` tem a função de verificar o correto funcionamento dos componentes do AFIS e disparar alertas de erro em caso de eventuais problemas. O componente `Audit Service` permite que sejam realizadas funções de auditoria sobre o AFIS, como identificar os responsáveis pela realização de ações específicas no sistema. O `Performance Monitor` é responsável por monitorar as métricas do sistema (e.g., uso de recursos, taxa de requisições) e prover relatórios estatísticos sobre as consultas. As informações são persistidas no `InfluxDB`, um banco de dados específico para dados em séries temporais.

O `User Manager` é responsável por gerenciar os usuários do AFIS (e.g., adicionar e remover), bem como alterar suas permissões de uso do sistema. O `Configuration Manager` gerencia os diversos parâmetros de configuração do AFIS, como o número de requisições simultâneas que podem ser tratadas, quantidade de `Workers` ativos, regras de escalabilidade, configurações específicas do `Template DB` e `Indexing Engine`.

O `Enrollment Worker` é responsável por cadastrar, atualizar ou remover impressões digitais de indivíduos no AFIS. Para a remoção, o componente recebe o identificador dos indivíduos que devem ser removidos do sistema. Esta identificação não é de responsabilidade do AFIS, permitindo a associação com bases de dados de usuários já existentes. Para inclusão ou atualização, o `Enrollment Worker` avalia a qualidade das imagens que recebe antes de sua inclusão no sistema, pois aquelas com baixa qualidade (limiar definido a partir da configuração do AFIS) serão recusadas e o cliente notificado. O `Enrollment Worker` gera o *template* (i.e., conjunto de minúcias) a partir da imagem, ao invocar um dos componentes de extração (`Minutiae Extractor`), responsável por processar e extrair as minúcias. Ressalta-se que podem haver diferentes versões dos extratores, utilizando diferentes abordagens e algoritmos para extrair as minúcias das digitais e geração de respectivos *templates* de representação. Após obter o *template*, o `Enrollment Worker` cuidará de toda a lógica de inclusão junto ao `Indexing Engine`.

O `Template DB Redis` é o componente responsável por incluir, atualizar e remover *templates* (as informações de minúcias dos indivíduos) da base, de forma segura e replicável. O `Template DB Redis` também é responsável por manter o componente `Indexing Engine` atualizado ao sincronizar inserções/atualizações e remoções de indivíduos. O `Indexing Engine` é responsável por fazer a indexação das informações biométricas, a fim de possibilitar rápidas consultas à base de digitais. Ao receber uma impressão digital para consulta, este componente deve retornar uma lista ranqueada com as digitais mais similares. O tamanho da lista de retorno é configurável no AFIS. O `Identification Worker` é o componente que solicita ao `Indexing Engine` os *templates* mais similares ao enviado na consulta. Os *templates* inicialmente são carregados do Redis e alocados em memória. O `Identification Worker` então utiliza *templates* em memória para melhorar o desempenho das comparações pelo `Matcher` e calcular a similaridade do *template* de consulta. Os resultados da indexação são comparados com os *templates* em memória para melhoria de desempenho.

Em caso de verificação, um identificador do indivíduo é enviado junto com a imagem ou *template* de consulta. O `Verification Worker` solicita ao `Template DB Redis` os *templates* do indivíduo a ser verificado e então executa o `Matcher` para calcular a similaridade do *template*

de consulta e os *templates* retornados pelo `Template DB Redis`.

O componente `LatentWorker` permite que sejam realizadas consultas de identificação em digitais latentes, que levam mais tempo por serem comparadas com um conjunto maior da base. O `Indexing Engine` encapsula o *ElasticSearch* para indexar os *templates* dos indivíduos e fornecer dados para o componente de identificação. Este componente guarda as informações de indexação no banco `MCCVectorDB`. O `SubjectDB` permite armazenar os indivíduos inseridos no sistema, bem como suas imagens. O `LatentDB` armazena os resultados da consulta latente, bem como eventuais erros.

IV. Experimentação

Com o intuito de validar a arquitetura proposta, o AFIS foi implementado utilizando de microsserviços, fazendo uso de diversas ferramentas *open-source*, estando de acordo com o estado de arte de aplicações de grande porte, elásticas e robustas. Foi utilizado o *service broker* RabbitMQ como *Load Balancer* de requisições, o algoritmo utilizado para avaliação de qualidade das digitais foi o NFIQ2 [10], as extrações de minúcias e posteriormente o matching de digitais foram feitos através do extrator e matching SourceAFIS. As minúcias extraídas armazenadas no banco Redis, para armazenamento e indexação das impressões digitais foi utilizada a ferramenta ElasticcSearch como MCCVectorDB. Informações recebidas pelos Enrollmentwork sobre os indivíduos cadastrados foram armazenados SubjectDB, utilizando o PostgresSQL.

Para avaliar o desempenho do AFIS foram testadas as features de inserção de indivíduos no sistema, de verificação e identificação, e internamente na identificação foi testado o mecanismo de indexação. Foi disponibilizado uma base de dados real com 10 mil indivíduos pela instituição, cada indivíduo com 10 imagens de dedos, e as informações de identificador único e sexo, e as imagens tinham o formato WSQ (Wavelet Scalar Quantization) um tipo de arquivo comprimido que facilita no transporte de imagens na rede.

Em todos os testes foi utilizada uma máquina com processador Intel(R) Xeon(R) CPU E5645 @ 2.40GHz, com 24 threads, 32 GB de memória e sistema operacional Ubuntu 18.04. Neste ambiente foi instalado o Kubernetes versão 1.15. Nos testes realizados, apenas o MCCVector DB - Elastic Search foi executado fora do ambiente Kubernetes pois tal configuração apresentou maior desempenho. As funcionalidades complexas do sistemas foram testadas de forma sistemática e detalhadas ao decorrer desta seção.

A. Avaliação de Inserção de Indivíduos

Neste primeiro conjunto de testes foi avaliada a capacidade do sistema de cadastrar novos indivíduos de forma paralela. Para isso, foram incluídos 10 mil indivíduos da base disponibilizada, onde foi garantido que cada um destes indivíduos fosse diferente dos demais, tendo identificação única.

Foram disponibilizadas 10 réplicas *EnrollmentWorker*, e consequentemente 10 réplicas de WebServices para cadastro. Cada réplica do WebService recebia requisições via *HTTP* e encaminhava as requisições para os workers de inclusão via *RabbitMQ*. Cada réplica executava com uma única *thread*, realizando cada etapa de cadastro do indivíduo de forma sequencial.

Para processar sequencialmente os procedimentos que compõem uma única inserção de indivíduo, cada réplica levou, em média, um total de $25,84$ segundos desde a chegada da requisição ao *EnrollmentWorker* via *RabbitMQ* até o momento que ele envia a resposta ao próprio *RabbitMQ*. Para melhor detalhamento do tempo total gasto, o processo de inserção foi dividido em diferentes etapas sequenciais. São elas:

- **Decodificação** Nesta etapa, cada imagem recebida é decodificada de base64 para um vetor binário. Em seguida, tal vetor é convertido para matriz *OpenCV*, onde é também verificado se a imagem está no formato *WSQ*;
- **Validação da qualidade NFIQ2**: Cada imagem tem sua qualidade testada utilizando o método *NFIQ2*, que prediz quão boa uma imagem é para ser armazenada e ser utilizada em futuras verificações. Imagens com qualidade inferior ao limiar previamente definido são rejeitadas e retiradas dos registros do AFIS. Nos testes realizados, o valor *NFIQ2* utilizado para validação foi 20. Devido a este filtro, aproximadamente $10,5\%$ das imagens foram descartadas;
- **Extração**: As imagens com boa qualidade detectadas na etapa de *NFIQ2* são submetidas ao processo de extração de minúcias, tanto para armazenamento de templates quanto para o processo de indexação, onde são utilizados os extratores *Mindtct* e *SourceAFIS*;
- **Armazenamento no Redis**: Os templates gerados pelo *SourceAfis* gerados na etapa de extração são armazenados no Redis para futuros processos de verificação e identificação.
- **Armazenamento no Postgres**: Todas as informações de cada indivíduo inserido, além de suas imagens na forma base64, são armazenadas no *Postgres* para eventuais consultas por parte do usuário;
- **Envio para Indexação**: Os templates coletados na etapa de extração utilizando o *Mindtct* são transformados em vetores binários e enviados para a *IndexingEngine* e aguarda confirmação de recebimento.

Durante os testes, cada procedimento foi medido em milissegundos (ms), e teve calculada sua média, cujos resultados são apresentados na Tabela I.

Etapa	Tempo médio em ms (σ)
Decodificação	$689,52$ $(140,60)$
Validação NFIQ2	$15.323,00$ $(3.562,00)$
Extração de Template	$6.273,20$ $(2.014,30)$
Armazenamento no Redis	$11,00$ $(3,70)$
Armazenamento no Postgres	58.60 $(23,50)$
Envio à Indexing Engine	$475,00$ $(138,70)$
Média geral por Indivíduo	$25.842,00$ $(3.022,00)$

Tabela I
Tempos médios (ms) durante teste de inclusão de indivíduos.

Com a utilização de diversas réplicas foi possível inserir

diversos indivíduos ao mesmo tempo. Como cada réplica usa apenas uma *thread*, possibilitou inserir paralelamente diversos indivíduos de acordo com as possibilidades da máquina, onde todos levaram igualmente a média total mostrada na Tabela I.

Dos 10 mil indivíduos inseridos, cada um com 10 impressões digitais únicas, foram cadastrados 89509 impressões digitais com nível aceitável de qualidade, levando todo o experimento 7,17 horas (com as 10 réplicas mencionadas) para sua execução. Todos os demais testes foram executados utilizando a base de 89509 digitais.

B. Avaliação da Identificação

Neste teste, também foi utilizada a base com 10 mil indivíduos, onde no procedimento de identificação, o AFIS retorna um conjunto de digitais que, representam as digitais mais similares à digital fornecida. Nos testes realizados, o AFIS foi configurado para retornar 2000, 4000, 6000, 8000 e 10000 digitais, e com 1, 4, 8, 12 e 24 *threads*. Foram realizadas 30 consultas de identificação para cada combinação de *threads*, número de resultados da indexação e filtros (dedo, mão e sexo), totalizando 6000 consultas.

1) Desempenho da Identificação: A Figura 2 apresenta os resultados dos testes de desempenho da identificação realizados com o AFIS. É importante frisar que nestes testes, o tempo de verificação corresponde a uma busca em um conjunto com os identificadores dos indivíduos retornados, onde já se sabe de antemão, qual o identificador do indivíduo procurado.

2) Qualidade da Indexação: Neste experimento, para cada identificação requisitada, buscou-se no vetor de retorno se a digital correspondente encontrava-se ou não. Foi feita uma comparação 1:1 (*matching*) entre a digital fornecida e cada digital retornada no conjunto. Este teste permitiu verificar a qualidade do processo de indexação realizado. O objetivo foi determinar o percentual das requisições em que o indivíduo esperado é devolvido pela indexação. Com isso, foi possível identificar se, quando uma identificação falha, o problema é com o *matching* ou se o indivíduo não foi considerado por não ter sido devolvido pela indexação.

Foi usado o cadastro da base de 89 mil digitais, e indexadas no *Elastic Search*. Nos testes foram feitas requisições com os seguintes parâmetros: filtros de mão, dedo e sexo, por fim sem filtros, e com com 500, 1000 e 2000 resultados requisitados. Para cada caso foram feitas 50 requisições, totalizando 600 requisições. As requisições utilizaram segundas vias de digitais já existentes na base. Os resultados são apresentados na Tabela II: os filtros melhoram significativamente os resultados da indexação. Isso se deve, principalmente, à redução do tamanho do espaço de buscas quando utilizamos filtros.

3) Precisão da Identificação: Neste experimento, a partir do conjunto de digitais retornado, foi realizado o procedimento de *matching* entre a digital procurada e cada digital do conjunto. Foi calculado a porcentagem de acerto da identificação usando a base de 10 mil indivíduos inseridos pelo *Enrollment Worker*. Esses testes foram feitos com filtros (mão, dedo e sexo) e sem filtros. Assim como no teste anterior, foram considerados casos em que o retorno da indexação fornecia

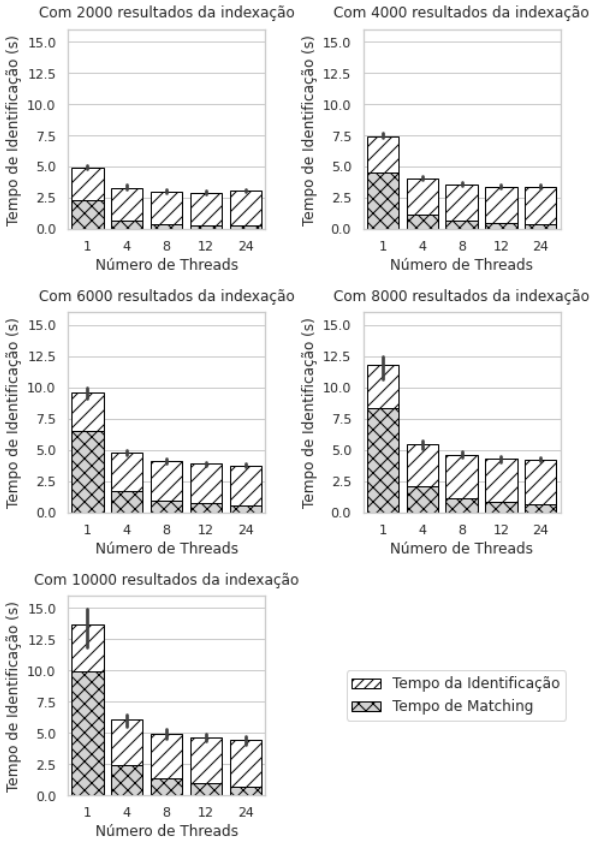

Figura 2. Resultados dos testes de desempenho de identificação

Resultados	Acertos	% Acerto	Filtros		
			Dedo	Mão	Sexo
500	20	40%	Não	Não	Não
1000	21	42%	Não	Não	Não
2000	23	46%	Não	Não	Não
500	21	42%	Não	Sim	Não
1000	23	46%	Não	Sim	Não
2000	25	50%	Não	Sim	Não
500	26	52%	Sim	Sim	Não
1000	32	64%	Sim	Sim	Não
2000	35	70%	Sim	Sim	Não
500	32	64%	Sim	Sim	Sim
1000	35	70%	Sim	Sim	Sim
2000	42	84%	Sim	Sim	Sim

Tabela II
RESULTADOS DO EXPERIMENTO DE INDEXAÇÃO (SEM *matching*)

2000, 4000, 6000, 8000 e 10000 resultados. Para cada caso foram feitas 30 requisições, totalizando 300 requisições. Os resultados são apresentados na Figura 3.

Conforme esperado, os testes indicam que o fornecimento de informações adicionais é fundamental para a melhoria da precisão do AFIS.

C. Avaliação da Verificação

A verificação realiza o procedimento de *matching* da digital passada com uma digital que está inserida no banco, correspondente ao id, dedo e mão que foi passado. Após o

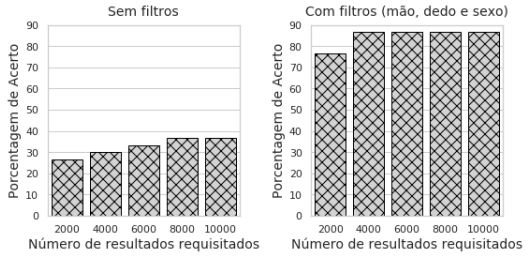

Figura 3. Resultados de acerto da identificação

procedimento é retornado um *score*. Caso esse *score* seja maior ou igual ao limiar escolhido, a digital é considerada correspondente, caso contrário, a digital passada não corresponde a que está no banco de dados.

1) Desempenho da Verificação: Para a execução do teste de desempenho foi utilizado a base de digitais disponibilizada pela instituição. Durante o processo foram selecionados 1000 indivíduos, com exatamente 10 dedos cada e 2 imagens por dedo (primeiras e segundas vias), totalizando 20 mil imagens. As imagens da primeira via foram inseridas no banco de dados e as segundas vias serviram para realização do teste de verificação de digitais. O tempo médio para verificação de cada indivíduo foi de $0{,}87$ segundos.

2) Validação da Verificação: Para o teste de validação da verificação foi feita a comparação entre genuínos e impostores. Uma comparação entre genuínos significa que será feita uma verificação entre duas digitais pertencentes ao mesmo indivíduo, mão e dedo. Por outro lado, uma comparação entre impostores é a verificação entre duas digitais da mesma mão e do mesmo dedo, porém de indivíduos diferentes. Utilizamos a segunda via das digitais dos 1000 indivíduos da base disponibilizada pela instituição e aplicamos as métricas de Verdadeiro Positivo (VP), Verdadeiro Negativo (VN), Falso Positivo (FP), Falso Negativo (FN), False Match Rate (FMR), False Non-Match Rate (FNMR) e Acurácia (Acc).

No teste realizado, o limiar 80 foi escolhido por ser o padrão para o *SourceAFIS*. Utilizando esse limiar na base de testes de verificação, podemos observar na Tabela III que foi obtido uma probabilidade de 0% de aparição de falsos positivos, em contrapartida, a probabilidade de aparecer um falso negativo é de 44% e com uma taxa de acerto de 77%.

Limiar	VP	VN	FP	FN	FMR	FNMR	Acc
80	530	10000	0	4470	0	0,447	0,7765
36	6906	9999	1	3094	0,0001	0,3094	0,8452

Tabela III
VERIFICAÇÃO COM DOIS DIFERENTES LIMIARES E 20 MIL DIGITAIS

Utilizando uma mesma base, a variação do limiar implica em diferentes resultados: quanto maior o limiar, menor e maior serão as chances de aparecerem falsos positivos e falsos negativos, respectivamente; quanto *menor* o limiar, maior a chance de falsos positivos e menor a chance de falsos negativos. Foram então realizados testes com uma base fornecida pela instituição com as vias de 16 mil indivíduos para identificar o limiar a ser utilizado. Foram realizados mais de 2 milhões de *matchings* entre digitais, considerando quatro valores de limiares: 36, 40, 54 e 64. O Limiar 36 foi aquele obteve os melhores resultados, com uma taxa de acurácia de $99,84\%$, FMR de $0,02\%$ e FNMR de $14,22\%$. Entretanto, nesse teste com 1000 indivíduos, esse limiar obteve uma probabilidade de $0,01\%$ de aparecer um falso positivo, 30% de surgir um falso negativo e obteve uma taxa de acerto de 84% (ver Tabela III).

V. CONCLUSÕES E TRABALHOS FUTUROS

O uso de biometria digital para identificação e verificação de pessoas é fortemente atrelada aos Sistemas Automáticos de Identificação de Impressões Digitais. O AFIS aqui proposto foi desenvolvido a partir de tecnologias abertas e estruturado de forma a dar suporte a um grande número de indivíduos, referentes à população.

Os experimentos realizados sugerem que o AFIS possui desempenho promissor e suporte adequado a uma grande demanda de usuários. Como trabalhos futuros, pretende-se incluir informações biométricas de face na arquitetura, além de melhorar a extração [11] e o método de indexação via *deep learning* para melhorar os desempenhos de identificação [12].

AGRADECIMENTOS

Os autores agradecem à Fundação Cearense de Apoio ao Desenvolvimento Científico e Tecnológico (FUNCAP) pelo financiamento do projeto (6945087/2019).

REFERÊNCIAS

[1] S. Liu and M. Silverman, "A practical guide to biometric security technology," *IT Professional*, vol. 3, no. 1, pp. 27–32, 2001.

[2] D. Peralta, I. Triguero, R. Sanchez-Reillo, F. Herrera, and J. M. Benítez, "Fast fingerprint identification for large databases," *Pattern Recognition*, vol. 47, no. 2, pp. 588–602, 2014.

[3] Y.-x. Zhao, W.-x. Zhang, D.-s. Li, Z. Huang, M.-n. Li, and X.-c. Lu, "Pegasus: a distributed and load-balancing fingerprint identification system," *Frontiers of Information Technology & Electronic Engineering*, vol. 17, no. 8, pp. 766–780, 2016.

[4] J. Xu, J. Jiang, Y. Dou, and X. Shen, "A low-cost fully pipelined architecture for fingerprint matching," in *2014 12th International Conference on Signal Processing (ICSP)*. IEEE, 2014, pp. 413–418.

[5] M. Lastra, J. Carabaño, P. D. Gutiérrez, J. M. Benítez, and F. Herrera, "Fast fingerprint identification using gpus," *Information Sciences*, vol. 301, pp. 195–214, 2015.

[6] R. M. Jiang and D. Crookes, "Fpga-based minutia matching for biometric fingerprint image database retrieval," *Journal of Real-Time Image Processing*, vol. 3, no. 3, pp. 177–182, 2008.

[7] C. Watson, "Nist special database 14-mated fingerprint card pairs 2," *National Institute of Standards and Technology*, 1993.

[8] R. Cappelli, D. Maio, and D. Maltoni, "Synthetic fingerprint-database generation," in *Object recognition supported by user interaction for service robots*, vol. 3. IEEE, 2002, pp. 744–747.

[9] R. Cappelli, M. Ferrara, and D. Maltoni, "Large-scale fingerprint identification on gpu," *Information Sciences*, vol. 306, pp. 1–20, 2015.

[10] Federal Office for Information Security, "NIST fingerprint image quality 2.0," National Institute of Standards and Technology, NIST Interagency Report, April 2016.

[11] A. G. Medeiros, J. P. B. Andrade, P. B. S. Serafim, A. M. M. Santos, J. G. R. Maia, F. A. M. Trinta, J. A. F. de Macêdo, P. P. R. Filho, and P. A. L. Rego, "A novel approach for automatic enhancement of fingerprint images via deep transfer learning," in *2020 International Joint Conference on Neural Networks (IJCNN)*, 2020, pp. 1–8.

[12] L. F. da Costa, L. S. Fernandes, J. P. B. Andrade, P. A. L. Rego, and J. G. R. Maia, "Deep convolutional features for fingerprint indexing," in *Intelligent Systems*. Cham: Springer International Publishing, 2021, pp. 223–237.