

**UNIVERSIDADE PAULISTA  
CIÊNCIA DA COMPUTAÇÃO**

**DIOGO RAMOS LOPES DA SILVEIRA – C173398**

**RICARDO FERNANDES SOUTO – C13CHI5**

**WANDERSON MARTINS OLIVEIRA – C202754**

**YURY RODRIGUES ANUNCIAÇÃO – C203360**

**DESENVOLVIMENTO DE UM SISTEMA DE IDENTIFICAÇÃO E AUTENTICAÇÃO  
BIOMÉTRICA**

**SÃO PAULO  
2016**

## SUMÁRIO

<b>1 OBJETIVOS.....</b>	<b>2</b>
<b>1.1 Geral.....</b>	<b>2</b>
<b>1.2 Específicos.....</b>	<b>2</b>
<b>2 INTRODUÇÃO.....</b>	<b>3</b>
<b>3 FUNDAMENTOS DAS PRINCIPAIS TÉCNICAS BIOMÉTRICAS.....</b>	<b>5</b>
<b>3.1 Características físicas.....</b>	<b>6</b>
<b>3.1.1 Impressões digitais.....</b>	<b>6</b>
<b>3.1.2 Reconhecimento da íris.....</b>	<b>8</b>
<b>3.1.3 Reconhecimento facial.....</b>	<b>8</b>
<b>3.1.4 Geometria das veias.....</b>	<b>9</b>
<b>3.2 Características comportamentais.....</b>	<b>9</b>
<b>3.2.1 Padrão de digitação.....</b>	<b>9</b>
<b>3.2.2 Reconhecimento de assinaturas.....</b>	<b>10</b>
<b>3.2.3 Reconhecimento de voz.....</b>	<b>11</b>
<b>3.3 Classificação dos sistemas biométricos.....</b>	<b>11</b>
<b>4 PLANO DE DESENVOLVIMENTO DA APLICAÇÃO.....</b>	<b>13</b>
<b>4.1 Ambiente de desenvolvimento.....</b>	<b>14</b>
<b>4.2 Interface gráfica.....</b>	<b>15</b>
<b>4.3 Leitor Biométrico.....</b>	<b>16</b>
<b>4.4 Computador e Sistema Operacional.....</b>	<b>17</b>
<b>4.5 FFV SDK.....</b>	<b>17</b>
<b>4.6 Banco de dados.....</b>	<b>18</b>
<b>4.7 Imagens.....</b>	<b>19</b>
<b>5 PROJETO DO PROGRAMA.....</b>	<b>20</b>
<b>5.1 janelaCadastroUsuarios.....</b>	<b>20</b>
<b>5.1.1 Principal.....</b>	<b>20</b>
<b>5.1.2 LoginAdminGUI.....</b>	<b>21</b>
<b>5.1.3 LoginAdminCtrl.....</b>	<b>21</b>
<b>5.1.4 JanelaGUI.....</b>	<b>21</b>
<b>5.1.5 JanelaCtrl.....</b>	<b>22</b>
<b>5.2 janelaAcessoDB.....</b>	<b>23</b>
<b>5.2.1 Principal.....</b>	<b>23</b>

<b>5.2.2 LoginDBGUI.....</b>	<b>23</b>
<b>5.2.3 LoginDBCtrl.....</b>	<b>24</b>
<b>5.2.4 JanelaGUI.....</b>	<b>24</b>
<b>5.3 compartilhada.....</b>	<b>25</b>
<b>5.3.1 Usuario.....</b>	<b>25</b>
<b>5.3.2 JdialogProgressoLeituraDigital.....</b>	<b>26</b>
<b>5.3.3 ScannerNffv.....</b>	<b>26</b>
<b>5.3.4 TrataErrosExcecaoEscaner.....</b>	<b>26</b>
<b>5.3.5 SobreGUI.....</b>	<b>26</b>
<b>6 RELATÓRIO COM AS LINHAS DE CÓDIGO DO PROGRAMA.....</b>	<b>27</b>
<b>7 APRESENTAÇÃO DO SISTEMA EM FUNCIONAMENTO EM UM COMPUTADOR.....</b>	<b>37</b>
<b>REFERÊNCIAS.....</b>	<b>51</b>

## 1 OBJETIVOS

### 1.1 Geral

Desenvolver, utilizando a linguagem de programação Java, um sistema de identificação e autenticação biométrica com interface gráfica, para restringir o acesso dos usuários a um determinado banco de dados. Esta restrição deve ser feita levando-se em consideração os níveis de permissão de acesso que os usuários têm há determinadas informações, que serão divididas em 3 níveis.

O banco de dados que deve ser protegido pertence ao Ministério do Meio Ambiente. Contendo informações sigilosas sobre as propriedades rurais que fazem uso de agrotóxicos proibidos, causando grandes impactos ambientais.

### 1.2 Específicos

São objetivos específicos deste trabalho:

- Pesquisar e dissertar sobre os conceitos gerais da biometria;
- Definir o tipo de autenticação biométrica que será utilizada;
- Definir a estrutura do sistema;
- Criar a interface gráfica;
- Implementar o projeto utilizando a linguagem de programação Java.

## 2 INTRODUÇÃO

Com a cada vez maior informatização da sociedade e a transferência de informações para o mundo digital, é de vital importância conseguir proteger estes dados. Os métodos tradicionais de proteção, que utilizam um nome de usuário e senha para impedir acessos não autorizados, tem demonstrado ser insuficientes nesta tarefa, pois sua segurança é muito baixa.

O principal tópico de segurança para proteger serviços de rede ou alguma determinada área do sistema, é a autenticação e identificação dos usuários. Se esta etapa não conseguir impedir acessos indevidos, então dificilmente outro método de segurança implementado terá eficiência nesta tarefa.

É neste cenário onde a biometria surge como um dos principais, mais eficazes, seguros e confiáveis métodos de identificação e autenticação indivíduos.

Biometria (palavra derivada do grego *bio*, que significa vida, e *metria*, que significa medição) é a medição e a análise estatística das características físicas e comportamentais dos indivíduos. A premissa básica da autenticação biométrica é que todos são únicos e que um indivíduo pode ser identificado por suas características físicas ou traços comportamentais. A tecnologia tem sido usada fortemente como o principal meio de controle de acesso e/ou para identificação de indivíduos sob vigilância.

O uso da impressão digital para assinar documentos foi pioneiro no conceito de biometria, sendo utilizada por diversos povos em diferentes épocas, desde os antigos babilônios, egípcios, assírios, japoneses e chineses. Os mesmos utilizam a impressão digital para marcar seus produtos e lacrar documentos. A impressão digital ajudava a identificar bons e maus negociantes com os quais os mercadores já haviam realizado algum tipo de negócio. Os chineses costumavam com o uso de papel e tinta, marcar pés e mãos das crianças para diferenciá-las, podendo ser considerado um dos primeiros exemplos de uso da certidão de nascimento.

Os avanços comerciais e de automatizações relacionadas à utilização da biometria tiveram início na década de setenta. Nesta época, surgiu um sistema conhecido como Identimat, esse sistema foi instalado em vários locais secretos para controle de acesso. Funcionava medindo a palma da mão e analisando o tamanho dos dedos. O Identimat teve sua produção encerrada na década de oitenta, mas seu pioneirismo foi importante na divulgação e aceitação da tecnologia biométrica

perante a população.

Junto ao desenvolvimento da tecnologia baseada na mão, a biometria baseada em impressão digital começa a ganhar maior espaço. Durante esse período, algumas empresas estavam trabalhando com a identificação automática das impressões digitais para ajudar as forças policiais. A comparação das imagens das digitais armazenadas em registros criminais, era, até então, realizada manualmente o que demandava muito tempo e esforço.

Outros métodos foram criados e os antigos continuaram a ser melhorados, enquanto a adoção dos baseados impressões digitais avançava. O primeiro sistema a analisar o padrão único da retina iniciou sua utilização na metade dos anos oitenta. No mesmo período na Universidade de Cambridge, foi desenvolvida a tecnologia baseada na íris.

No Brasil em 2004 foi inaugurado a AFIS(*Automated Fingerprint Identification System*, ou, Sistema Automatizado de Identificação de Impressões Digitais), que foi interligado com o sistema de Informações Criminais (Sinic) criando o Sistema de Identificação Nacional (SIN). O SIN foi inaugurado com 800 mil impressões digitais de criminosos.

Em 2007 o governo brasileiro iniciou a emissão do passaporte biométrico. Contendo diversos itens de segurança, o novo sistema de passaporte coleta assinatura, foto e dez impressões digitais roladas.

Nos últimos anos está sendo implementada a utilização da biometria em eleições, com o intuito de aumentar a segurança do processo eleitoral, buscando evitar fraudes. Muitos países têm desenvolvido grandes inovações nesta área, tanto nos processos de expedição dos documentos eleitorais quanto nos procedimentos da própria votação em si. O maior exemplo são as urnas eletrônicos com autenticação biométrica.

### 3 FUNDAMENTOS DAS PRINCIPAIS TÉCNICAS BIOMÉTRICAS

A autenticação e/ou identificação de um determinado indivíduo pode ser feita de diversas formas. Podemos classificá-las em três grandes grupos: por aquilo que se possui, por aquilo que se sabe e por aquilo que se é..

**Por que se sabe:** a autenticação é realizada utilizando algo que o usuário conhece. Exemplos são os nomes de acesso(nomes de usuário), senhas e chaves criptográficas. Em termos de segurança, são os de nível mais baixo.

**Por aquilo que se possui:** estes métodos realizam a autenticação baseada em algo que o usuário possui. Geralmente são utilizados dispositivos que contém memória e/ou capacidade de processamento. Os exemplos mais conhecidos são os *Tokens*, *Smart Cards*(cartões bancários, cartão de identidade pessoal, “chip” de celulares), cartões magnéticos e crachás. Estes métodos possuem nível de segurança média.

**Por aquilo que você é:** a autenticação é baseada naquilo que o usuário é. Ou seja, é nesta categoria que se encaixa a biometria. Podendo ser utilizada uma característica física ou uma característica comportamental única, que permita identificar e distinguir um indivíduo do outro. Exemplos de características biométricas são a impressão digital, a íris, a retina, a voz, a assinatura, etc. São considerados os métodos que oferecem o mais alto nível de segurança.

O emprego da biométrica em tecnologias de autenticação já faz parte do dia a dia da maior parte da população. Seu uso mais visível são os caixas eletrônicos dos bancos(que estão adotando este método) e as urnas eletrônicas utilizadas nas votações eleitorais.

Também é utilizada como método de verificação de funcionários, validação de associados de planos de saúde, controle de acesso em locais de entrada restrita, segurança de redes, comércio eletrônico, acesso virtual, catraca eletrônica, etc. A lista de usos somente tende a aumentar conforme a sociedade automatiza e informatiza seus processos.

Um sistema de autenticação biométrica é dividido em duas fases: o registro do perfil do usuário e a sua autenticação. Na primeira fase é realizado o cadastramento da amostra biométrica dos usuários. Isto é feito utilizando-se um dispositivo que permita a captura destas características biométricas, como, por exemplo, leitor de digital, microfone, câmera de vídeo, etc.

Após a captura, a amostra é transformada em um algoritmo matemático que será armazenado em um banco de dados. Toda vez que for preciso autenticar um usuário, será capturada uma nova amostra biométrica que será comparada com o modelo que está armazenado.

Com relação aos métodos utilizados na identificação biométrica, eles podem ser divididos em dois grandes grupos: aqueles que utilizam características físicas e aqueles que utilizam características comportamentais.

### **3.1 Características físicas**

Utilizam a forma ou composição do corpo como identificador único dos indivíduos.

#### **3.1.1 Impressões digitais**

Método automatizado utilizado para a identificação ou confirmação de indivíduo baseado em uma comparação entre dois dedo. É o tipo mais conhecido e usado de biometria atualmente. As razões para sua popularidade são sua facilidade de aquisição, uso estabelecido, a sua aceitação em comparação com os demais e o fato de que há dez fontes biométricas por indivíduo.

A impressão digital é formada por diversas linhas, criadas a partir das elevações da pele. Ela é única para cada indivíduo, mesmo irmãos gêmeos possuem impressões digitais diferentes. Sendo justamente por isto, um dos métodos de identificação mais seguros existentes. Tendo sido descoberto e utilizado a mais de mil anos.

Esta característica individual sofre poucas alterações durante a vida de uma pessoa. Alguns fatores externos podem modificá-la, como por exemplo, o uso de produtos químicos ou trabalhos manuais que acabem danificando as linhas da impressão digital.

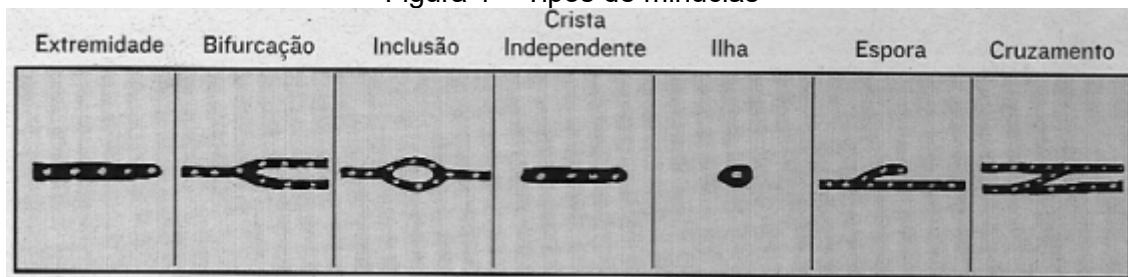
A identificação biométrica utilizando este método, funciona através da extração dos pontos de minúcias da impressão digital. Após a extração estes valores são processados realizando-se diversos cálculos a partir dos quais os sistemas computacionais identificam o indivíduo dono da impressão digital.

Minúcias é nome dado ao conjunto das principais características de uma

impressão digital. Na Figura 1 são mostrados exemplos de minúcias.

Existem três padrões básicos da impressão digital: o arco, o laço e a espiral. Algumas imagens de impressões digitais que apresentam estes padrões podem ser vistas na Figura 2.

Figura 1 – Tipos de minúcias



Fonte: GTA – Grupo de Teleinformática e Automação da UFRJ

Figura 2 – Padrões básicos da impressão digital: laço, arco e espiral



Fonte: Biometric Solutions(adaptada pelos autores)

Um arco é um padrão onde a linha da digital entra em um dos lados do dedo, cresce ao chegar no meio formando um arco e sai pelo outro lado. No laço, mais famoso padrão de digital, a linha entra por um dos lados, forma uma curva e sai pelo mesmo local pelo qual entrou. Já a espiral é um padrão onde são formados círculos ao redor de um ponto central.

Com relação aos dispositivos utilizados na coleta da impressão digital, eles são de três tipos: ótico, capacitivo e ultrassônico. O primeiro, e o mais utilizado, trabalha com uma fonte emissora de luz e utiliza a reflexão da luz sobre o dedo para realizar a leitura da digital. Os dispositivos capacitivos medem o calor emitido pela digital. E o último, o ultrassônico, como o próprio nome indica, envia sinais sonoros e coleta a impressão digital analisando o sinal retornado. Ou seja, funciona como um radar.

### 3.1.2 Reconhecimento da íris

Método automatizado utilizado para identificação ou confirmação de um indivíduo analisando padrões aleatórios da íris. O reconhecimento por íris é relativamente novo, sendo somente comercialmente desenvolvido na última década principalmente devido a limitações de patente.

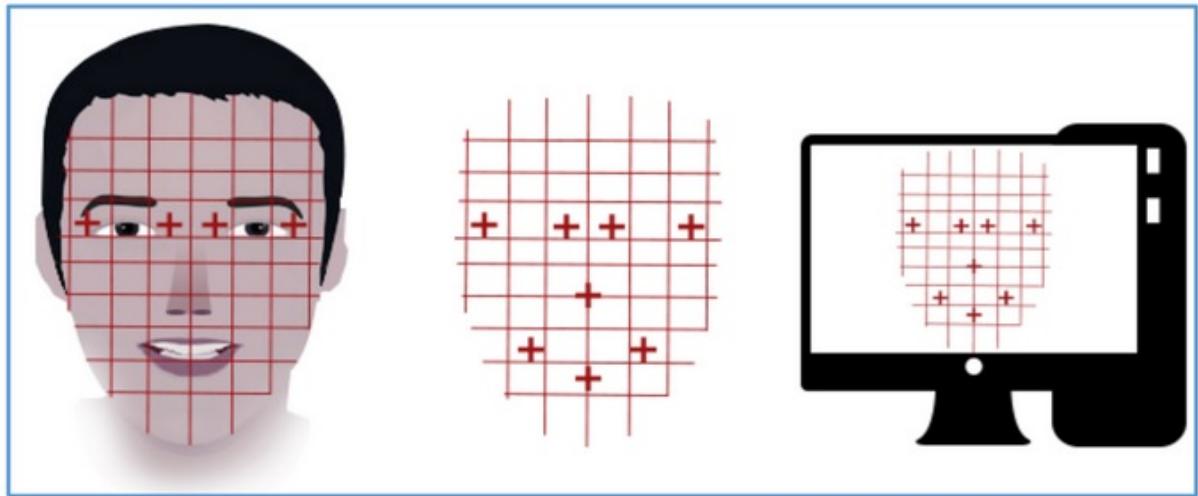
A íris é um músculo dentro do olho que regula o tamanho da pupila, controlando a quantidade de luz que entra no olho. A íris é a parte colorida do olho e cada uma possui sua própria estrutura, o que permite sua utilização na identificação e autenticação de indivíduos.

Esta característica individual não se modifica com o passar dos anos, mantendo-se a mesma por toda a vida.

### 3.1.3 Reconhecimento facial

O reconhecimento fácil utiliza a geometria espacial de diferentes características da face (confira a Figura 3). Estas características não são modificadas, mesmo que se realize uma cirurgia plástica. Alguns exemplos de medidas são a distância entre os olhos; a distância entre os olhos, nariz e boca; e a distância entre as linhas dos cabelos, os olhos, a boca e o queixo.

Figura 3 – Reconhecimento facial



Fonte: Tutorials Point

Uma grande diferença entre o reconhecimento facial e os demais métodos é

que o reconhecimento facial pode ser capturado à distância, como, por exemplo, por câmera de seguranças, sendo possível sua aplicação sem o conhecimento do sujeito. Sendo adequado para encontrar crianças perdidas e procurar criminosos.

É um dos métodos menos intrusivos, pois não é necessário utilizar informações consideradas extremamente pessoais, como a impressão digital ou a composição da íris.

Outra vantagem é que pode-se utilizar, em teoria, quaisquer câmeras digitais como dispositivo de captura da amostra biométrica. Bastando que se tenha um programa de identificação e autenticação biométrica que suporte aquela câmera.

### **3.1.4 Geometria das veias**

Tal como as impressões digitais, o padrão das veias corporais são diferentes entre os indivíduos, sendo um excelente critério para a identificação biométrica. O reconhecimento é confirmado a partir dos vasos sanguíneos que ficam na superfície da pele. Geralmente a parte do corpo utilizada para este método são as mãos.

Atualmente, os dois maiores exemplos de instituições que utilizam este método são o FBI e a CIA, ambas dos Estados Unidos da América.

## **3.2 Características comportamentais**

Utilizam o comportamento como identificador único dos indivíduos.

### **3.2.1 Padrão de digitação**

Método utilizado para a identificação usando como critério os padrões de digitação do indivíduo. As medidas utilizadas por este critério são tempo de permanência (duração que uma tecla é pressionada) e o tempo de não-permanência (duração entre o soltar de uma tecla e o pressionar de outra).

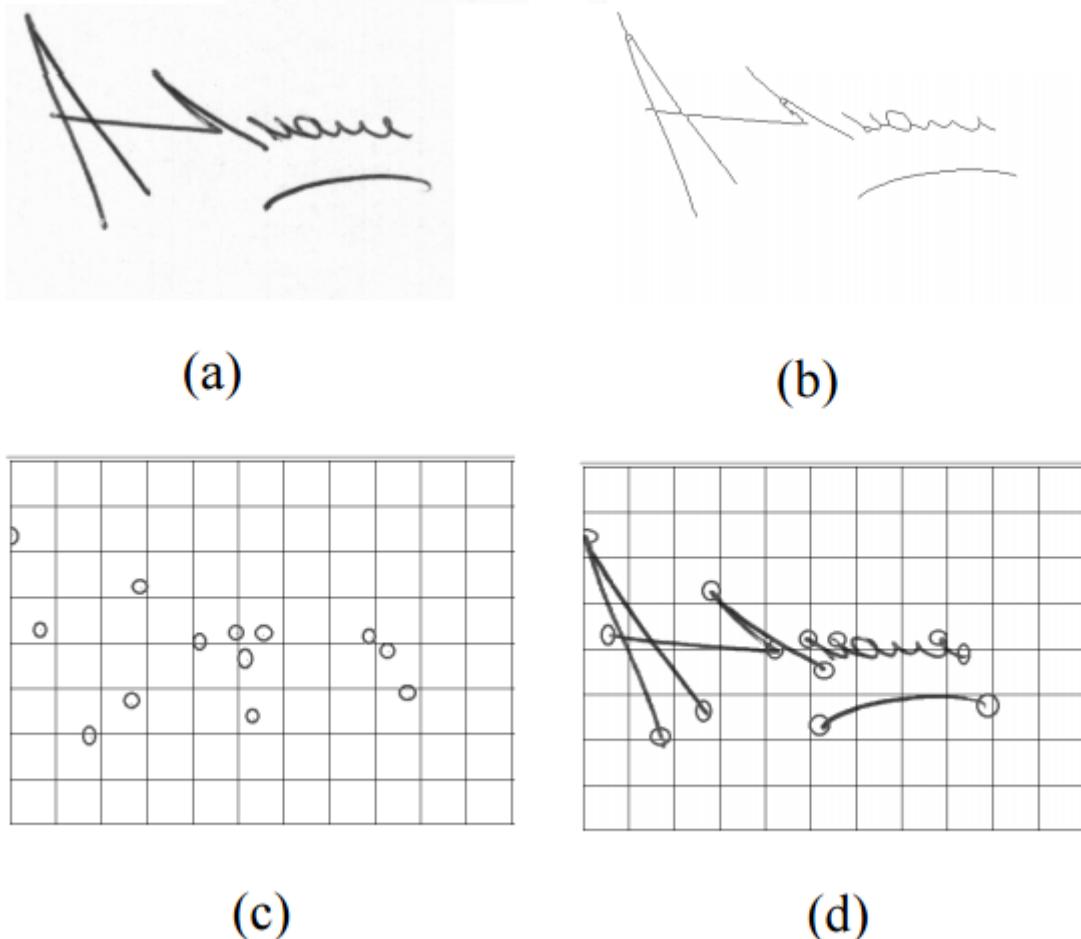
Este padrão foi utilizado na segunda guerra mundial e ficou conhecido como *The Fist of the Sender* (O punho do remetente). Tendo sido utilizado pelas agências de inteligência militar para distinguir, baseado no ritmo de escrita, se um código morse teria sido enviado por um aliado ou inimigo de guerra.

Como atualmente, na maior parte das residências existe ao menos um teclado, este tipo de método biométrico é o mais fácil de ser implementado, se considerarmos a questão de aquisição do dispositivo de leitura biométrica.

### 3.2.2 Reconhecimento de assinaturas

Método de baixo nível que leva como critério a assinatura pessoal que cada pessoa opta por utilizar como meio de identificação pessoal. Costuma ser muito confundida com os autógrafos, sendo estes utilizados por artistas e pessoas públicas. Geralmente grandes artistas possuem uma assinatura, onde a mantém privada, e um autógrafo que é utilizado de forma pública.

Figura 4 – Primitivas de “início e fim abruptos”. Imagem original(a), imagem esqueletizada(b), pontos de início e fim do segmento no grid(c), pontos de início e fim na imagem original(d)



Fonte: GTA – Grupo de Teleinformática e Automação da UFRJ(adaptada pelos autores)

O reconhecimento de assinaturas é o método mais utilizado na comprovação de documentos, principalmente em Bancos e cartórios. Ele utiliza diversas características para realizar a análise a assinatura. Entre elas pode-se citar a velocidade, a pressão, a direção e o sentido da escrita..

Na Figura 4 está exemplificado parte da análise automatizada de uma digital.

### 3.2.3 Reconhecimento de voz

Refere-se ao método automatizado para reconhecimento da identidade de um indivíduo baseado em sua voz. A sua principal desvantagem reside no fato de que nem todo ambiente pode implementar este método. Pois quanto maior a poluição sonora menor será a precisão do reconhecimento da voz. Outra desvantagem é que o estado de saúde do indivíduo, como gripe ou estresse, pode dificultar sua identificação.

A voz possui dois fatores biométricos, fisiológico e comportamental:

- O componente fisiológico da fala é a forma física da voz do sujeito.
- O componente comportamental trata-se do movimento da mandíbula, lábios, língua, etc.

Existem dois tipos de reconhecimento de voz:

- Dependente de texto: O sujeito tem que pronunciar uma frase fixa (senha de acesso) que é a mesma tanto para o registro quanto para a verificação.
- Independente de texto: Baseado em qualquer palavra dita pelo sujeito.

O tipo dependente de texto tem uma melhor performance, porém o outro método é mais flexível, podendo ser utilizado até mesmo com indivíduos que não queiram cooperar, como é o caso criminosos.

## 3.3 Classificação dos sistemas biométricos

Os sistemas biométricos podem ser classificados e comparados através das características humanas que eles utilizam e também em relação ao próprio conjunto de dispositivos e processos necessários para o seu funcionamento. A seguir estão algumas categorias de classificação.

**Universalidade:** Está relacionada ao fato de todas as pessoas serem dotadas da característica que será utilizada na identificação individual. Nem sempre a

característica será universal. A impressão digital, por exemplo, é uma das que mais pontuam neste quesito, pois o número de indivíduos que não possuem nenhum dedo na mão é irrisório. No sentido oposto, a biometria através da forma como uma pessoa anda, não é universal, pois a medição não pode ser realizada em portadores de deficiência física. Além de ser muito dependente do estado de saúde do indivíduo.

**Unicidade:** Mede o quanto a característica se diferencia entre duas pessoas. Deve-se buscar métodos onde a probabilidade de duas pessoas ter a mesma medida da característica seja extremamente baixa.

**Permanência:** Indica se a característica varia com o tempo. Por exemplo, mesmo com o envelhecimento, o DNA é uma medida que dificilmente sofre alterações. Já a voz, é mais suscetível de sofrer alterações durante a vida e por causa de doenças(o mais comum seria o resfriado).

**Coletabilidade:** Está relacionada as etapas do processo biométrico, indicando o quanto de tempo e esforço são necessários para sua execução. A autenticação biométrica baseada em DNA é extremamente demorada, o que é uma grande desvantagem deste método. Já a impressão digital pode ser rapidamente obtida, sem grande esforço.

**Performance:** É a medida do custo do dispositivo, do processamento e da quantidade de tempo utilizada para realizar a autenticação dos indivíduos.

**Precisão:** Com que exatidão o sistema biométrico consegue diferenciar os indivíduos.

**Aceitabilidade:** Refere-se ao quão bem o sistema biométrico foi aceito pelos seus usuários. Diversas características devem ser levadas em consideração para aumentar a aceitação, sendo as principais a privacidade e o conforto dos usuários.

**Proteção:** Define o nível de segurança do sistema. Ou seja, o quão difícil é enganar o processo de autenticação

## 4 PLANO DE DESENVOLVIMENTO DA APLICAÇÃO

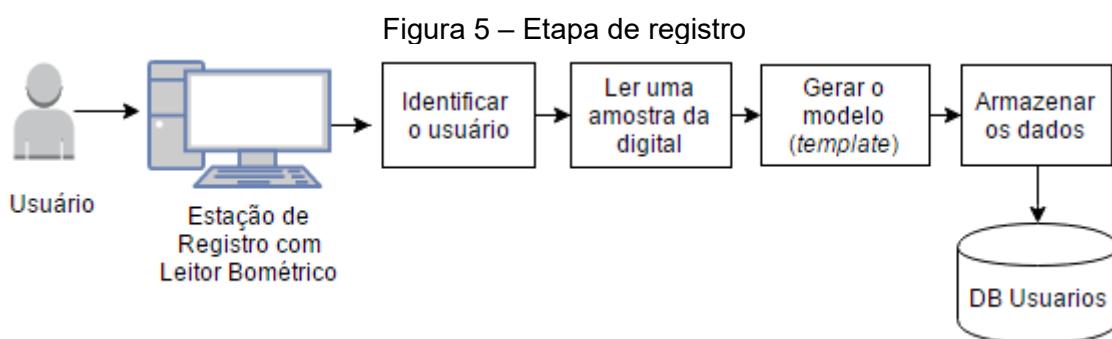
Neste projeto será desenvolvido um sistema de identificação e autenticação biométrica através da impressão digital. Combinando a utilização de *nome de usuário* com a biometria, fazendo uma comparação 1:1 na autenticação.

Serão desenvolvidas interfaces gráficas para intermediar todos os processos que dependem da interação dos usuários. Simples e diretas, focando nas funções principais da aplicação.

O sistema será formado por dois programas, o *Gerenciador de Usuários*, que cadastrará os usuários e alterará seus dados, e uma aplicação cliente, *Banco de Dados – MMA*, que implementará a autenticação biométrica para restringir o acesso aos dados sigilosos do Ministério do Meio Ambiente.

Para atingir esta finalidade será utilizada uma combinação de hardware e software. Na parte do hardware, o principal dispositivo será o leitor de impressão digital. Na parte lógica teremos, além da verificação da biometria, o uso de *nome de usuário* que identificará individualmente cada funcionário que terá acesso ao banco de dados.

Esta combinação entre a portabilidade do identificador do usuário e a complexidade e sofisticação da biometria aumentará a segurança das informações restritas. Garantindo que cada usuário possa acessar somente os recursos para os quais têm autorização.

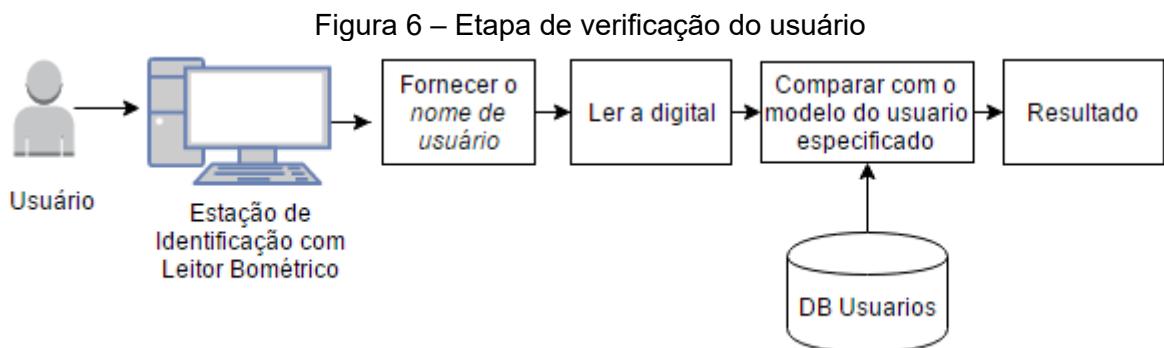


Fonte: Elaborada pelos autores

O sistema biométrico trabalha com o reconhecimento de padrões, adquirindo as informações biométricas(através do leitor de digital), gerando um modelo(*template*) a partir destes dados e armazenando este modelo para posteriormente ser utilizado para comparação e verificação do usuário. A Figura 5

demonstra estes processos.

Na fase de verificação os processos são basicamente os mesmos: primeiro é realizada uma leitura da digital do usuário, gerado um modelo e em seguida ele é comparado com o modelo previamente armazenado, para checar a real identidade do funcionário. Estes processos podem ser vistos na Figura 6.



Fonte: Elaborada pelos autores

Estas duas etapas têm nomes específicos. O procedimento de registro do perfil do usuário é conhecido como *enrollment*, e o de comparação é nomeado *matching*.

A autenticação dos usuários pode ser realizada de duas formas: 1:1 e 1:N. O primeiro método utiliza somente a impressão digital. Ele identifica o usuário comparando o modelo da impressão digital coletada com todos os modelos salvos no banco de dados. O que, dependendo da quantidade de usuários cadastrados, consumirá muitos recursos do sistema e demorará mais tempo para concluir a verificação.

O segundo método, 1:1, é o que será utilizado nesta aplicação. Ele realiza a verificação do usuário utilizando também um outro dado armazenado, que neste caso é o *nome de usuário*. Desta forma o modelo da impressão digital coletada será comparado somente com o modelo armazenado para o *nome de usuário* especificado. O que deixará a verificação mais rápida, principalmente quando se tem centenas de funcionários cadastrados, como é o caso Ministério do Meio Ambiente.

#### 4.1 Ambiente de desenvolvimento

O sistema será desenvolvido na linguagem de programação Java utilizando o

paradigma de Orientação a Objetos. Como IDE(*Integrated Development Environment*) foi utilizada o **Eclipse**(na versão Mars) junto com a **JRE na versão 1.8.**

Para realizar o controle das versões do sistema foi empregado o GIT. Fazendo uso do *plugin EGit* para adicionar suponte a esta ferramenta ao Eclipse. Permitindo que todos os membros da equipe possam manipular os arquivos do projeto simultaneamente, sem que estas alterações sobrescrevam as de outro membro. Garantido maior produtividade, velocidade e segurança no desenvolvimento do projeto.

Além deste *plugin*, também foi utilizado o **ObjectAid** no Eclipse. Através desta ferramenta é possível gerar diagramas e arquivos UML do projeto. Sendo assim, todas as figuras destes tipos, apresentadas neste trabalho, foram geradas pelo ObjectAid.

O **GIMP** foi o editor de imagens utilizado na manipulação das figuras deste sistema. Ele é um programa multiplataforma, *open source* e contém as principais funcionalidades e características do Photoshop, além das suas próprias ferramentas. Um de seus diferenciais em relação ao programa da Adobe é ser gratuito.

Figura 7 – Logos dos programas e *plugins* utilizados



Fonte: Das respectivas empresas e projetos.

#### 4.2 Interface gráfica

A interface gráfica será desenvolvida fazendo uso da biblioteca gráfica **Swing**. Todas as JRE, a partir da 1.2, trazem esta biblioteca como o padrão gráfico. Sua principal vantagem reside no fato de ser multiplataforma. Permitindo que as interfaces possam ser utilizadas em qualquer sistema operacional que suporte a JRE, apresentando os componentes da janela com os mesmos atributos(cores, tamanhos, margens, espaçamento, etc), independente da plataforma em que está sendo executada.

#### 4.3 Leitor Biométrico

Para a realização deste projeto foi empregado o leitor biométrico da DigitalPersona, o DigitalPersona U.are.U 4000 Sensor. Na Figura 8 está uma imagem do dispositivo e na Tabela 1 algumas de suas características.

Figura 8 – DigitalPersona U.are.U 4000 Sensor



Fonte: Amazon.com

Tabela 1 – Características do leitor biométrico

<b>DigitalPersona U.are.U 4000 Sensor</b>	
Tipo	Ótico
Conexão	USB 2.0
Resolução	512 DPI
Tamanho da imagem	355x390 pixels
SO suportado	Windows (32-bit e 64-bit), Linux (32-bit e 64-bit)

Fonte: Elaborada pelos autores

Este modelo é extremamente prático, leve e de fácil transporte. Tem uma

baixa taxa de falhas, produz imagens de excelente qualidade e é vendido por um preço abaixo dos concorrentes da mesma categoria. Sendo que o dispositivo adquirido para este projeto custou R\$ 200.

Ele foi desenvolvido, originalmente, para ser utilizado em computadores pessoais. Principalmente na substituição das senhas utilizadas na autenticação do usuário do sistema operacional.

#### **4.4 Computador e Sistema Operacional**

As configurações do notebook utilizado neste projeto, tanto para o desenvolvimento do sistema quanto em seu teste, estão listas na Tabela 2.

Tabela 2 – Configuração do computador

<b>Notebook Dell Inspiron 1440</b>	
Processador	Pentium® Dual-Core CPU T4300 @ 2.10GHz
Memória:	4GB
HD	Fujitsu 5400rpm de 300GB
Monitor	14.1 polegadas
Sistema Operacional	Windows 10 Pro 32bits

Fonte: Elaborada pelos autores

O sistema operacional escolhido foi o da Microsoft, pelo fato do *driver* do leitor biométrico disponibilizado pelo SDK funcionar apenas no Windows.

#### **4.5 FFV SDK**

SDK, *Software Development Kit*, é um kit de desenvolvimento para auxiliar no desenvolvimento de aplicações, neste caso, controlando o leitor biométrico e realizando as operações de cadastro, verificação e validação das impressões digitais de forma rápida e eficaz.

A DigitalPersona, fabricante do leitor biométrico disponibiliza seu próprio SDK, o U.are.U SDK. A desvantagem é que ele não é gratuito, precisando adquirir uma licença para utilizar. Não sendo disponibilizado sequer uma versão de testes.

Por este motivo, o SDK escolhido para utilizar neste projeto foi o **FFV SDK**, disponibilizado pela NeuroTechnology. Que suporta mais de 140 modelos de leitores

biométrico.

O Free Fingerprint Verification SDK é uma versão simplificado do VeriFinger SDK. Sendo indicado para aplicações que realizam *logon* biométrico. Sua limitação é o suporte a no máximo 10 *templates* de impressões digitais e permitir somente a autenticação 1:1.

O VeriFinger SDK é uma versão comercial desenvolvida pela mesma empresa(NeuroTechnology). Sem limitação na quantidade de *templates*, suporta autenticação 1:N e verificação digital através de arquivos(sem uso do leitor biométrico). A principal diferença prática para este projeto entre o FFV SDK e o VeriFinger SDK está no fato do primeiro somente armazenar os *templates* em um banco de dados proprietário da NeuroTechnology. Enquanto que o VeriFinger SDK permite o armazenamento em qualquer tipo de banco de dados.

O SDK comercial disponibiliza uma versão de teste de 30 dias. No entanto, para ela funcionar é preciso que a aplicação esteja conectada a internet durante seu uso. O que é uma desvantagem, pois nem sempre o acesso à internet estará garantido, o que impossibilitará o uso da aplicação. Por causa disto, foi decidido pela utilização do FFV SDK. Futuramente, pode ser feita uma migração para o VeriFinger SDK.

#### **4.6 Banco de dados**

Como o SDK utilizado no sistema somente permite a armazenamento dos *templates* no banco de dados proprietário da Neurotechnology, este será o utilizado. Basicamente, os modelos serão salvas em um arquivo criptografado.

Além dos *templates* será preciso armazenar os dados dos usuários, como seu nome, seu nome de usuário e seu nível de acesso. Isto será realizado através do salvamento do objeto do usuário em um arquivo. Este arquivo será armazenado em formato binário, utilizando a funcionalidade de Serialização disponibilizada pelo Java.

Desta forma o sistema terá dois bancos de dados, um contendo os *templates* e outro os dados dos usuários. Os dois serão arquivos e estarão armazenadas na pasta do sistema. Desta forma torna extremamente fácil mover a aplicação de um computador para outro, ou até mesmo, no futuro, centralizar o banco de dados em uma única máquina que terá conexão com várias aplicações de cadastro e validação de usuário.

#### 4.7 Imagens

Foi desenvolvida uma logomarca muito simples para ser utilizada na aplicação. Ela pode ser vista na Figura 9.

Outras duas imagens são utilizadas como ícones das janelas de diálogo. Uma para indicar que o usuário foi verificado com sucesso e outra para indicar que o usuário não foi verificado. Elas podem ser vistas na Figura 10.

Figura 9 – Logomarca da aplicação



Fonte: Elaborada pelos autores

Figura 10 – Ícones de verificação do usuário



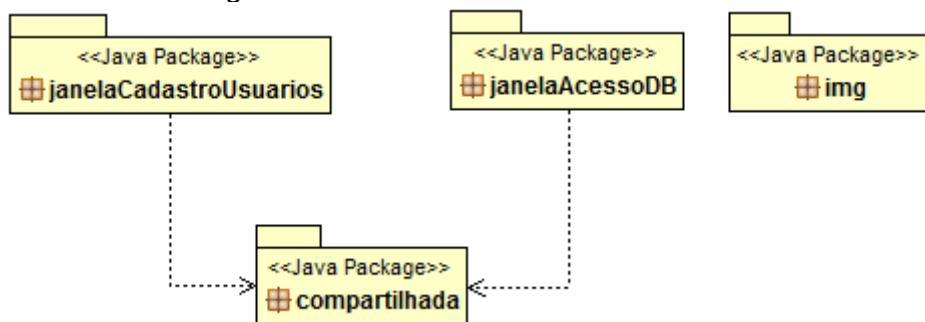
Fonte: Elaborada pelos autores

## 5 PROJETO DO PROGRAMA

Os arquivos do sistema foram organizados em 4 pacotes: **janelaCadastroUsuarios**, **janelaAcessoDB**, **compartilhada** e **img**. Como visto na Figura 11.

O pacote **janelaCadastroUsuarios** contém as classes da janela de cadastro dos usuários. No pacote **janelaAcessoDB** estão armazenadas as classes da janela que permite o acesso aos arquivos do banco de dados do Ministério do Meio Ambiente, sendo esta a janela que contém uma tela de verificação da identidade do usuário. Por sua vez, o último pacote, **img**, como o próprio nome indica, contém as imagens utilizadas no sistema.

Figura 11 – Pacotes utilizados no sistema



Fonte: Elaborada pelos autores

### 5.1 janelaCadastroUsuarios

A janela de cadastro dos usuários é formada por 6 classes: **JanelaCtrl**, **JanelaGUI**, **LoginAdminController**, **LoginAdminGUI**, **Principal** e **Usuario**.

#### 5.1.1 Principal

Esta é a principal classe deste pacote. Ela instancia um objeto do tipo **Nffv**, através da classe **ScannerNffv** do pacote **compartilhada**. Este objeto pertence ao SDK FFV e será utilizado para manipular o leitor biométrico e o banco de dados que armazenará os *templates* das impressões digitais.

Após esta instanciação, ela inicializa uma tela de autenticação do usuário administrador do banco de dados. Isto é feito através de um objeto do tipo

### ***LoginAdminGUI.***

Caso o resultado da autenticação seja positivo, é mostrada a tela de cadastro dos usuários(**JanelaGUI**).

#### **5.1.2 LoginAdminGUI**

Instancia o objeto controlador desta janela, **LoginAdminController** e exibe a tela para o administrador do banco de dados realizar sua autenticação. Isto é essencial, pois apenas o administrador pode ter acesso aos dados dos usuários cadastrados.

#### **5.1.3 LoginAdminController**

Responsável pelo controle da janela de autenticação do administrador. Verifica se já existe um administrador cadastrado, caso exista pede para ele realizar sua autenticação através da leitura de sua digital. Caso contrário, ou seja, se o banco de digitais estiver vazio, pede para o administrador cadastrar sua digital.

#### **5.1.4 JanelaGUI**

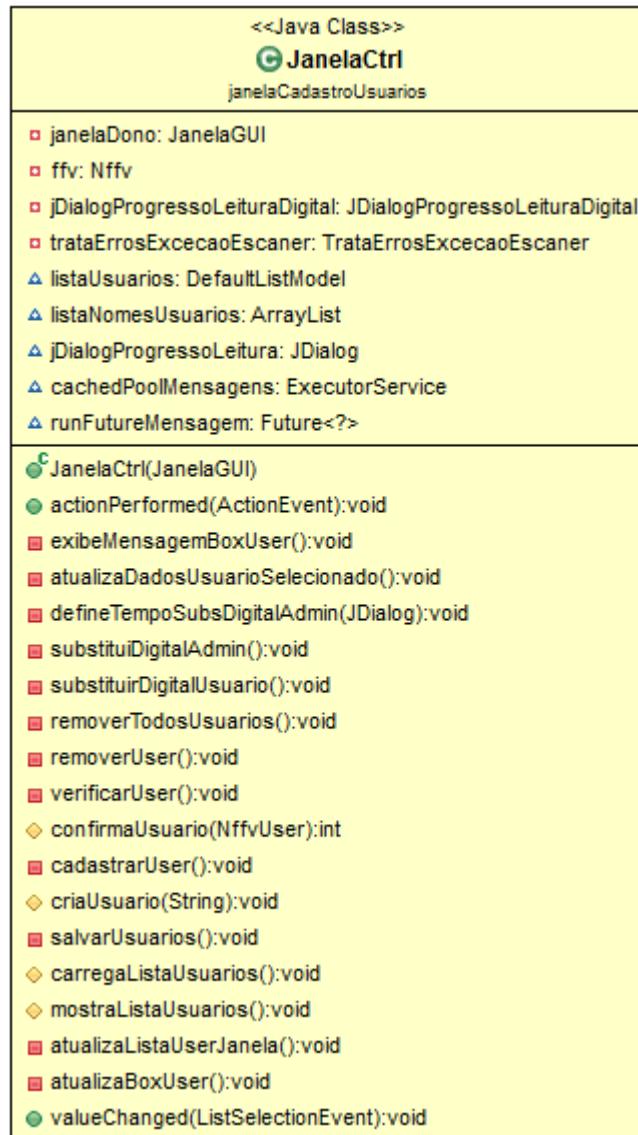
Exibe a janela de cadastro dos usuários e alteração de seus respectivos dados caso o administrador tenha sido autenticado com sucesso. Esta janela será controlada pela classe **JanelaCtrl**.

Nela será exibida a lista de usuários cadastrados e quando algum deles for selecionado seus dados serão mostrados no centro da janela. Permitindo que seja alterado o nome do funcionário, seu nome de usuário, seu nível de acesso e que sua digital armazenada possa ser substituída por outra.

A barra de menus da janela contém 5 opções: *Cadastrar*(permite o cadastro de um usuário no sistema), *Verificar*(verifica a digital do usuário, ou seja, compara a digital armazenada no banco de digitais com a que será lida), *Remover*(remove os usuários selecionados), *Remover todos*(remove todos os usuários cadastrados) e *Sobre*(exibe a janela que contém informações sobre o sistema e seus desenvolvedores).

### 5.1.5 JanelaCtrl

Figura 12 – Métodos e atributos da classe JanelaCtrl



Fonte: Elaborada pelos autores

Classe que contém a lógica por trás da aplicação de cadastro dos usuários. Na Figura 12 estão mostrados seus métodos e atributos. Abaixo segue a descrição de seus principais métodos.

- **exibeMensagemBoxUser**: exibe ou oculta mensagens referentes a alterações nos dados do usuário selecionado;
- **atualizaDadosUsuarioSelecionado**: responsável por salvar as alterações realizadas nos dados do usuário selecionado;
- **substituiDigitalAdmin**: substitui a digital do administrador do banco de dados;

- **substituirDigitalUsuario**: substitui a digital do usuário selecionado;
- **removerTodosUsuarios**: remove todos os usuários do banco de dados;
- **removerUser**: remove o usuário selecionado;
- **verificarUser**: verifica a digital do usuário selecionado. Pode ser utilizado, por exemplo, para confirmar a identidade do usuário antes de realizar alguma alteração em seus dados;
- **cadastrarUser**: cadastrá um usuário no sistema. Salvando sua digital no banco de digitais e seus dados em um banco de dados separado;
- **mostraListaUsuarios**: mostra na janela a lista de usuários cadastrados;
- **atualizaBoxUser**: atualiza os componentes da janela para exibir os dados do usuário selecionado.

## 5.2 janelaAcessoDB

Este pacote é da aplicação que permite acessar os dados sigilosos do MMA(Ministério do Meio Ambiente). Contendo ao todo 4 classes: **JanelaGUI**, **LoginDBCtr**, **LoginDBGUI** e **Principal**.

### 5.2.1 Principal

Realiza, basicamente as mesmas funções da classe **Principal** do pacote **janelaCadastroUsuarios**. Instancia um objeto do tipo **Nffv**, através da classe **ScannerNffv** do pacote **compartilhada**. Este objeto pertence ao SDK FFV e será utilizado para verificar o usuário que está tentando acessar o banco de dados do MMA.

Após isto, é inicializada uma tela de autenticação do usuário, através de um objeto do tipo **LoginDBGUI**. Caso a autenticação seja positiva, é mostrada a tela de acesso as informações salvas no banco de dados do Minitério do Meio Ambiente(**JanelaGUI**).

### 5.2.2 LoginDBGUI

Instancia o objeto controlador desta janela, **LoginDBCtr** e exibe a tela para o usuário realizar sua autenticação. Isto é essencial, pois apenas os usuários

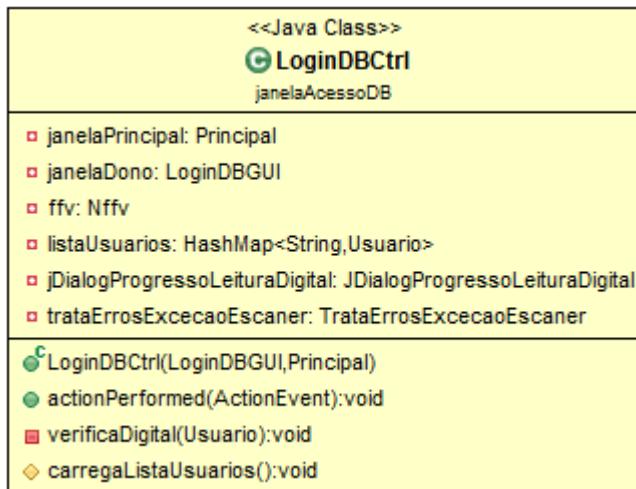
cadastrados podem ter acesso as informações confidenciais do Ministério do Meio Ambiente.

### 5.2.3 LoginDBCtrl

Responsável pelo controle da janela de autenticação do usuário. Pega o nome de usuário digitado, escaneia a digital e compara com o modelo armazenado no banco de digitais para o usuário especificado. Se o resultado da comparação for negativo informa para o usuário. Se for positivo, indica para a classe **Principal** que deve ser exibida a janela com os dados do Ministério do Meio Ambiente.

Na Figura 13 está sua apresentação em um diagrama UML.

Figura 13 – Métodos e atributos da classe LoginDBCtrl



Fonte: Elaborada pelos autores

### 5.2.4 JanelaGUI

Uma classe muito simples que mostra uma hipotética tela de acesso aos dados confidenciais no Ministério do Meio Ambiente. Dependendo do nível de acesso do usuário autenticado, será exibida uma mensagem informando quais tipos de arquivo ele pode acessar.

Os botões e menus de opções da janela não tem nenhuma funcionalidade, pois esta é uma tela de exemplo. O foco do sistema é a autenticação biométrica e não a criação de uma tela de acesso a um banco de dados.

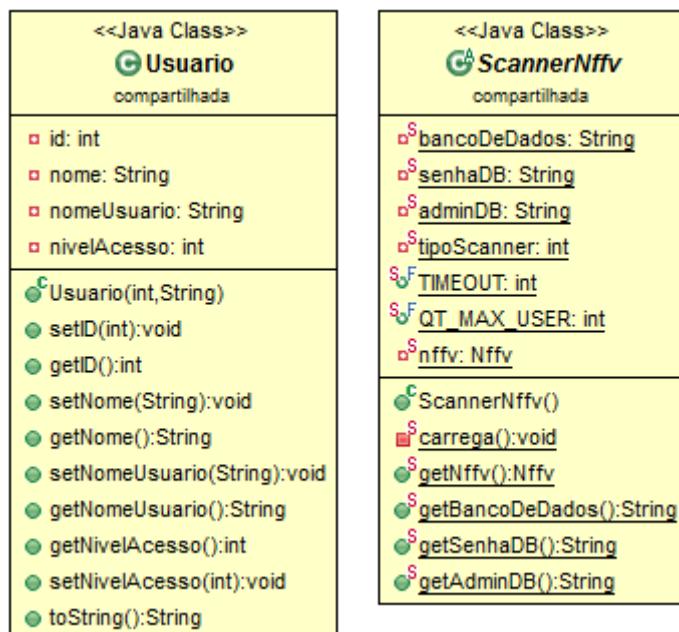
Ainda nesta janela, será exibido o nome do usuário e seu nível de acesso.

### 5.3 compartilhada

Este pacote contém as classes que são utilizadas tanto pela janela de cadastro dos usuários quanto pela janela de acesso ao banco de dados do MMA. Sendo formado pelas seguintes classes: **JdialogProgressoLeituraDigital**, **ScannerNffv**, **SobreGUI**, **TrataErrosExcecaoEscaner** e **Usuario**.

Na Figura 14 estão os métodos e atributos das classes **Usuario** e **ScannerNffv**.

Figura 14 – Métodos e atributos das classes Usuario e ScannerNffv



Fonte: Elaborada pelos autores

#### 5.3.1 Usuario

Classe que contém os dados do usuário(nome, nome de usuário e nível de acesso) além de seu identificador(id) que permite localizar sua digital que está armazenada no banco de digitais. Será utilizada tanto pela aplicação de cadastro dos usuários quanto pela aplicação que permite o acesso as informações do banco de dados do Ministério do Meio Ambiente.

Esta classe implementa a interface **Serializable**, permitindo que seus objetos sejam salvos em um arquivo que conterá todos os usuários cadastrados. Ou seja, os objetos desta classe serão salvos em um banco de dados extremamente simples.

### **5.3.2 JdialogProgressoLeituraDigital**

Responsável por criar e exibir uma janela de progresso, informando que está sendo aguardada a leitura da digital do usuário.

### **5.3.3 ScannerNffv**

Esta classe instancia um objeto do tipo **Nffv**, que será utilizado por todas as classes do sistema para manipular o leitor biométrico e o banco de dados que armazena os *templates* das impressões digitais.

Também estão definidos o nome do banco de digitais, a senha utilizada para ter acesso ao banco de dados, o nome do usuário administrador, o modelo de leitor biométrico utilizado, o tempo máximo de tentativa de leitura de uma digital, entre outras variáveis necessárias para o pleno funcionamento do sistema.

### **5.3.4 TrataErrosExcecaoEscaner**

Responsável pelo tratamento dos erros e exceções que ocorrerem durante a manipulação do leitor de digitais. Exemplos de erros são o *NoScanner*(nenhum leitor de digitais foi detectado) e o *ScannerTimeout*(quando o tempo máximo de tentativa de leitura de uma digital foi atingido).

### **5.3.5 SobreGUI**

Esta classe cria e exibe uma janela contendo informações sobre o sistema, sua versão, seus desenvolvedores e as bibliotecas utilizadas.

## 6 RELATÓRIO COM AS LINHAS DE CÓDIGO DO PROGRAMA

As próximas páginas contém os códigos fontes das principais classes do sistema. Partes dos códigos foram ocultadas, como, por exemplo, as linhas de importação de bibliotecas e classes(**import**).

Programa 1 – Classe JanelaCtrl do pacote janelaCadastroUsuarios

```
package janelaCadastroUsuarios;
public class JanelaCtrl implements ActionListener, ListSelectionListener {
    private JanelaGUI janelaDono;
    private Nffv ffv;
    private JDialo

```

```

        janelaDono.lblMsgDadosSalvos.setVisible(false);
    } catch (InterruptedException e){}}};
runFutureMensagem = cachedPoolMensagens.submit(runMensagem);}

private void atualizaDadosUsuarioSelecionado(){
    Usuario usuarioSelecionado =
(Usuario)janelaDono.listaUser.getSelectedItem();
    if(janelaDono.txtNome.getText().trim().length() == 0 ||
janelaDono.txtNomeUsuario.getText().trim().length() == 0){
        JOptionPane.showMessageDialog(janelaDono,"Preencha os campos Nome e
Usuário!","",JOptionPane.WARNING_MESSAGE); return;}
    if(!
usuarioSelecionado.getNomeUsuario().equals(janelaDono.txtNomeUsuario.getText())
&& listaNomesUsuarios.contains(janelaDono.txtNomeUsuario.getText()))
{
    JOptionPane.showMessageDialog(janelaDono,"O nome de usuário digitado
já estar sendo utilizado.", "",JOptionPane.WARNING_MESSAGE); return;}
    listaNomesUsuarios.remove(usuarioSelecionado.getNomeUsuario());
    listaNomesUsuarios.add(janelaDono.txtNomeUsuario.getText());
    usuarioSelecionado.setNome(janelaDono.txtNome.getText());

    usuarioSelecionado.setNomeUsuario(janelaDono.txtNomeUsuario.getText());
    usuarioSelecionado.setNivelAcesso((int)janelaDono.spinnerNivelAcesso.getValue());
    atualizaListaUserJanela();
    salvarUsuarios();
    janelaDono.lblMsgDadosSalvos.setText("Dados salvos com sucesso!");
    exibeMensagemBoxUser();}

private void defineTempoSubsDigitalAdmin(JDialog adminSubsDigital){
    if(cachedPoolMensagens == null){
        cachedPoolMensagens = Executors.newCachedThreadPool();}
    if(runFutureMensagem != null && !runFutureMensagem.isDone()){
        runFutureMensagem.cancel(true);}
    Runnable runMensagem = new Runnable(){ public void run(){ try{
        TimeUnit.SECONDS.sleep(60);
        adminSubsDigital.dispose();
        substituiDigitalAdmin();
    } catch (InterruptedException e){}}};
    runFutureMensagem = cachedPoolMensagens.submit(runMensagem);}

private void substituiDigitalAdmin(){
    JDialog jDialog = new JDialog(janelaDono, true);
    jDialog.setDefaultCloseOperation(JDialog.DISPOSE_ON_CLOSE);
    jDialog.setResizable(false);
    jDialog.setTitle("Banco de Dados - Ministério do Meio Ambiente");

    jDialog.setIconImage(Toolkit.getDefaultToolkit().getImage(LoginAdminGUI.class
.getResource("/img/icon-digital-verificada.png")));
    jDialog.setBounds(100, 100, 286, 138);
    jDialog.setLocationRelativeTo(janelaDono);
    jDialog.getContentPane().setLayout(new BorderLayout());
    JPanel contentPanel = new JPanel();
    contentPanel.setBorder(new EmptyBorder(5, 5, 5, 5));
    jDialog.getContentPane().add(contentPanel, BorderLayout.CENTER);
    contentPanel.setLayout(null);
    JLabel lblMsgInfo = new JLabel("É preciso verificar se você é o
Administrador:");
    lblMsgInfo.setFont(new Font("Dialog", Font.PLAIN, 12));
    lblMsgInfo.setBounds(12, 15, 256, 16);
    contentPanel.add(lblMsgInfo);
    JButton btnEscanearDigital = new JButton("escanear digital");
    btnEscanearDigital.setActionCommand("verificar");
}

```

```

btnEscanearDigital.setBounds(11, 51, 257, 33);
btnEscanearDigital.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if(e.getActionCommand().equals("verificar")){
            Usuario adminSelecionado =
(Usuario) listaUsuarios.getElementAt(0);
            NffvUser adminUsuario =
ffv.getUserByID(((Usuario)adminSelecionado).getID());
            int usuarioValidado = confirmaUsuario(adminUsuario);
            if(usuarioValidado == 1){
                lblMsgInfo.setText("Identidade verificada. Escanei a nova
digital:");
                lblMsgInfo.setForeground(new Color(34, 139, 34));
                btnEscanearDigital.setActionCommand("cadastrar");
                defineTempoSubsDigitalAdmin(jDialog);
            } else if(usuarioValidado == 0){
                ImageIcon imageIcon = new
ImageIcon(SobreGUI.class.getResource("/img/icon-digital-nao-
verificada.png"));
                Image image = imageIcon.getImage();
                Image novaImg = image.getScaledInstance(75, 63,
java.awt.Image.SCALE_SMOOTH);
                imageIcon = new ImageIcon(novaImg);
                JOptionPane.showMessageDialog(janelaDono,"As impressões
digitais não são compatíveis.", "Falha na verificação",
JOptionPane.ERROR_MESSAGE, imageIcon);
            }
        } else if(e.getActionCommand().equals("cadastrar")){
            substituirDigitalUsuario();
            jDialog.dispose();
        }
    });
    contentPanel.add(btnEscanearDigital);
    jDialog.setVisible(true);
}
private void substituirDigitalUsuario(){
try{
    SwingWorker<NffvUser,Void> worker = new SwingWorker<NffvUser,Void>(){
        protected NffvUser doInBackground(){
            return ffv.enroll(ScannerNffv.TIMEOUT);
        }
        protected void done(){DialogProgressoLeituraDigital.exibe(false);}
    };
    worker.execute();
    jDialogProgressoLeituraDigital.exibe(true);
    NffvUser novoUsuario = worker.get();
    if(ffv.getEngineStatus() != NffvStatus.TemplateCreated){
        trataErrosExcecaoEscaner.erro(ffv.getEngineStatus()); return;
    }
    Usuario usuarioSelecionado =
(Usuario)janelaDono.listaUser.getSelectedValue();
    ffv.removeUserID(usuarioSelecionado.getID());
    usuarioSelecionado.setID(novoUsuario.getID());
    salvarUsuarios();
    janelaDono.lblMsgDadosSalvos.setText("Digital substituída!");
    exibeMensagemBoxUser();
} catch (Exception e){trataErrosExcecaoEscaner.excecao(e);}

janelaDono.listaUser.setSelectedIndex(janelaDono.listaUser.getSelectedInde
x());
}
private void removerTodosUsuarios(){
Object[] options = { "SIM", "NÃO" };

```

```

    int desejaRemover = JOptionPane.showOptionDialog(janelaDono,"Deseja
remover todos os usuários cadastrados?", "",JOptionPane.YES_NO_OPTION,
JOptionPane.WARNING_MESSAGE,null,options,options[1]);
    if(desejaRemover == 1) return;
    for(int i=(listaUsuarios.size()-1); i>=0; i--) {

if(((Usuario)listaUsuarios.get(i)).getNome().equals(ScannerNffv.getAdminDB()
())){
    continue;
    ffv.removeUserID(((Usuario)listaUsuarios.get(i)).getID());
    listaUsuarios.removeElement((Usuario)listaUsuarios.get(i));
}
atualizaListaUserJanela();
salvarUsuarios();}

private void removerUser(){
Object[] usuariosSelecionados =
janelaDono.listaUser.getSelectedValues();
if(usuariosSelecionados.length == 0){
    JOptionPane.showMessageDialog(janelaDono,"Selecione o usuário que
deseja remover","",JOptionPane.ERROR_MESSAGE); return;}
Object[] options = { "SIM", "NÃO" };
int desejaRemover = JOptionPane.showOptionDialog(janelaDono,"Deseja
remover este(s) usuário(s)?","",JOptionPane.YES_NO_OPTION,
JOptionPane.WARNING_MESSAGE,null,options,options[1]);
if(desejaRemover == 1) return;
for (Object usuario : usuariosSelecionados) {
    if(((Usuario)usuario).getNome().equals(ScannerNffv.getAdminDB())){
        continue;
        ffv.removeUserID(((Usuario)usuario).getID());
        listaUsuarios.removeElement((Usuario)usuario);
        atualizaListaUserJanela();
        salvarUsuarios();
    }

janelaDono.listaUser.setSelectedIndex(janelaDono.listaUser.getFirstVisible
Index());}

private void verificarUser(){
Usuario usuarioSelecionado =
(Usuario)janelaDono.listaUser.getSelectedValue();
if(usuarioSelecionado == null){
    JOptionPane.showMessageDialog(janelaDono,"Selecione o usuário que
deseja verificar","",JOptionPane.ERROR_MESSAGE); return;}
NffvUser usuarioDB = ffv.getUserByID(usuarioSelecionado.getID());
int usuarioValidado = confirmaUsuario(usuarioDB);
if(usuarioValidado == 1){
    ImageIcon imageIcon = new
ImageIcon(SobreGUI.class.getResource("/img/icon-digital-verificada.png"));
    Image image = imageIcon.getImage();
    Image novaImg = image.getScaledInstance(85, 74,
java.awt.Image.SCALE_SMOOTH);
    imageIcon = new ImageIcon(novaImg);
    JOptionPane.showMessageDialog(janelaDono,
        usuarioSelecionado.getNome() + " foi verificado(a). \nAs
impressões digitais são compatíveis.", "Verificado",
        JOptionPane.DEFAULT_OPTION,imageIcon);}
else if(usuarioValidado == 0){
    ImageIcon imageIcon = new
ImageIcon(SobreGUI.class.getResource("/img/icon-digital-nao-
verificada.png"));
    Image image = imageIcon.getImage();
    Image novaImg = image.getScaledInstance(85, 74,

```

```

java.awt.Image.SCALE_SMOOTH);
    imageIcon = new ImageIcon(novaImg);
    JOptionPane.showMessageDialog(janelaDono,
        usuarioSelecionado.getNome() + " não foi verificado(a).\nAs
impressões digitais não são compatíveis.", "Falha na verificação",
        JOptionPane.ERROR_MESSAGE, imageIcon);}}
protected int confirmaUsuario(NffvUser usuario) {
    SwingWorker<Integer,Void> worker = new SwingWorker<Integer,Void>() {
        protected Integer doInBackground() {
            return ffv.verify(usuario, ScannerNffv.TIMEOUT);
        }
        protected void done(){jDialogProgressoLeituraDigital.exibe(false);}
    };
    worker.execute();
    jDialogProgressoLeituraDigital.exibe(true);
    int compatibilidadeUsuario = 0;
    try { compatibilidadeUsuario = worker.get();
    } catch (InterruptedException | ExecutionException e) {
        trataErrosExcecaoEscaner.excecao(e);
    }
    if (ffv.getEngineStatus() == NffvStatus.TemplateCreated) {
        if(compatibilidadeUsuario > 0){return 1;} else{return 0;}}
    else{
        trataErrosExcecaoEscaner.erro(ffv.getEngineStatus()); return -1;}}
private void cadastrarUser(){
    if(listaUsuarios.size() >= ScannerNffv.QT_MAX_USER){
        JOptionPane.showMessageDialog(janelaDono,"Só é permitido o cadastro
de 9 usuários :","",JOptionPane.ERROR_MESSAGE); return;}
    String nomeUsuario = JOptionPane.showInputDialog(janelaDono,
        new JLabel("Digite o nome do usuário"),"Cadastrar usuário",
        JOptionPane.QUESTION_MESSAGE);
    if(nomeUsuario == null) return;
    if(nomeUsuario.trim().length()==0){cadastrarUser(); return;}
    if(nomeUsuario.equals(ScannerNffv.getAdminDB())){
        JOptionPane.showMessageDialog(janelaDono,"O nome do usuário é
inválido!","",JOptionPane.ERROR_MESSAGE); cadastrarUser(); return;}
    criaUsuario(nomeUsuario);

janelaDono.listaUser.setSelectedIndex(janelaDono.listaUser.getLastVisibleI
ndex());}
protected void criaUsuario(String nomeUsuario){
try{
    SwingWorker<NffvUser(Void)> worker = new SwingWorker<NffvUser(Void)>()
{
    protected NffvUser doInBackground() {
        return ffv.enroll(ScannerNffv.TIMEOUT);
    }
    protected void done(){jDialogProgressoLeituraDigital.exibe(false);}
    };
    worker.execute();
    jDialogProgressoLeituraDigital.exibe(true);
    NffvUser novoUsuario = worker.get();
    if(ffv.getEngineStatus() != NffvStatus.TemplateCreated) {
        trataErrosExcecaoEscaner.erro(ffv.getEngineStatus()); return;}
    listaUsuarios.addElement(new
Usuario(novoUsuario.getID(),nomeUsuario));
    salvarUsuarios();
}catch (Exception e) {trataErrosExcecaoEscaner.excecao(e);}}
private void salvarUsuarios(){
    File arquivoDB = new File(ScannerNffv.getBancoDeDados() + ".fdb");
    try{
        FileOutputStream arquivoOut = new FileOutputStream(arquivoDB);

```

```

ObjectOutputStream arquivo      = new ObjectOutputStream(arquivoOut);
for (int i = 0; i < listaUsuarios.getSize(); i++){
    arquivo.writeObject(listaUsuarios.get(i));
}
arquivo.close();
} catch (Exception e) {e.printStackTrace();}
protected void carregaListaUsuarios(){
    listaUsuarios = new DefaultListModel();
    listaNomesUsuarios = new ArrayList();
    File arquivoDB = new File(ScannerNffv.getBancoDeDados() + ".fdb");
    if(arquivoDB.exists()){
        try{
            FileInputStream arquivoIn      = new FileInputStream(arquivoDB);
            ObjectInputStream arquivo      = new ObjectInputStream(arquivoIn);
            for (Usuario usuario = (Usuario)arquivo.readObject(); usuario != null; usuario = (Usuario)arquivo.readObject()){
                listaUsuarios.addElement(usuario);
                listaNomesUsuarios.add(usuario.getNomeUsuario());
            }
            arquivo.close();
        } catch (EOFException eof){}
        catch (Exception e) {e.printStackTrace();}
    }
}
protected void mostraListaUsuarios(){
    janelaDono.listaUser.setModel(listaUsuarios);
    janelaDono.listaUser.setSelectedIndex(0);
    atualizaBoxUser();
}
private void atualizaListaUserJanela(){janelaDono.listaUser.updateUI();}
private void atualizaBoxUser(){
    Usuario usuarioSelecionado =
    (Usuario)janelaDono.listaUser.getSelectedValue();
    if(usuarioSelecionado == null){
        janelaDono.txtNome.setText("");
        janelaDono.txtNomeUsuario.setText("");
        janelaDono.spinnerNivelAcesso.setValue(1);
        janelaDono.lblImgDigital.setIcon(null);
        janelaDono.txtNome.setEnabled(false);
        janelaDono.txtNomeUsuario.setEnabled(false);
        janelaDono.spinnerNivelAcesso.setEnabled(false);
        janelaDono.btnSalvarDadosUser.setEnabled(false);
        janelaDono.btnSubstituirDigitalUser.setEnabled(false);
        janelaDono.btnRemover.setEnabled(false);
        janelaDono.btnVerificar.setEnabled(false); return;}
    NffvUser usuario = ffv.getUserByID(usuarioSelecionado.getID());
    janelaDono.txtNome.setText(usuarioSelecionado.getNome());
    try {

janelaDono.lblImgDigital.setIcon(usuario.getNffvImage().getImageIcon());
    } catch (Exception e) {e.printStackTrace();}
    janelaDono.btnVerificar.setEnabled(true);
    janelaDono.txtNome.setEnabled(true);
    janelaDono.btnSubstituirDigitalUser.setEnabled(true);
    if(usuarioSelecionado.getNome().equals(ScannerNffv.getAdminDB())){
        janelaDono.txtNomeUsuario.setText("");
        janelaDono.spinnerNivelAcesso.setValue(1);
        janelaDono.txtNomeUsuario.setEnabled(false);
        janelaDono.spinnerNivelAcesso.setEnabled(false);
        janelaDono.btnSalvarDadosUser.setEnabled(false);
        janelaDono.btnRemover.setEnabled(false);}
    else{
}
}

```

```

janelaDono.txtNomeUsuario.setText(usuarioSelecionado.getNomeUsuario());

janelaDono.spinnerNivelAcesso.setValue(usuarioSelecionado.getNivelAcesso());
    janelaDono.txtNomeUsuario.setEnabled(true);
    janelaDono.spinnerNivelAcesso.setEnabled(true);
    janelaDono.btnSalvarDadosUser.setEnabled(true);
    janelaDono.btnRemover.setEnabled(true);
    if((listaUsuarios.size()-1) > 0){
        janelaDono.btnRemoverTodos.setEnabled(true);
    } else{ janelaDono.btnRemoverTodos.setEnabled(false); }
    public void valueChanged(ListSelectionEvent e) {atualizaBoxUser();}
}

```

Fonte: Elaborado pelos autores

### Programa 2 – Classe LoginDBCtrl

```

package janelaAcessosDB;
public class LoginDBCtrl implements ActionListener {
    private Principal janelaPrincipal;
    private LoginDBGUI janelaDono;
    private Nffv ffv;
    private HashMap<String, Usuario> listaUsuarios;
    private JDialogProgressoLeituraDigital jDialogProgressoLeituraDigital;
    private TrataErrosExcecaoEscaner trataErrosExcecaoEscaner;
    public LoginDBCtrl(LoginDBGUI janelaDono, Principal janelaPrincipal) {
        this.janelaDono = janelaDono;
        this.ffv = ScannerNffv.getNffv();
        this.janelaPrincipal = janelaPrincipal;
        carregaListaUsuarios();
        jDialogProgressoLeituraDigital = new JDialogProgressoLeituraDigital(this.janelaDono.getJDialog());
        trataErrosExcecaoEscaner = new TrataErrosExcecaoEscaner(this.janelaDono.getJDialog());
    }
    public void actionPerformed(ActionEvent e) {
        if(e.getSource() == janelaDono.btnEscanearDigital) {
            if(janelaDono.txtNomeUsuario.getText().length() == 0){
                JOptionPane.showMessageDialog(janelaDono.getJDialog(),
                    "Digite seu nome de usuário!", "", JOptionPane.WARNING_MESSAGE);
                return;
            } else{
                Usuario usuario =
                listaUsuarios.get(janelaDono.txtNomeUsuario.getText());
                if(usuario == null){
                    JOptionPane.showMessageDialog(janelaDono.getJDialog(), "O nome de
                    usuário é inválido", "", JOptionPane.ERROR_MESSAGE); return;
                } else{verificaDigital(usuario);}
            }
        }
    }
    private void verificaDigital(Usuario usuario){
        NffvUser usuarioDB = ffv.getUserByID(usuario.getID());
        SwingWorker<Integer,Void> worker = new SwingWorker<Integer,Void>(){
            protected Integer doInBackground(){
                return ffv.verify(usuarioDB, ScannerNffv.TIMEOUT);
            }
            protected void done(){jDialogProgressoLeituraDigital.exibe(false);}
        };
        worker.execute();
        jDialogProgressoLeituraDigital.exibe(true);
        int compatibilidadeUsuario = 0;
        try {

```

```

        compatibilidadeUsuario = worker.get();
    } catch (InterruptedException | ExecutionException e) {
        trataErrosExcecaoEscaner.excecao(e);
    if (ffv.getEngineStatus() == NffvStatus.TemplateCreated) {
        if( compatibilidadeUsuario > 0){
            janelaPrincipal.setUsuarioLogado(usuario);
            janelaDono.getJDialog().dispose();
        } else{
            ImageIcon imageIcon = new
ImageIcon(SobreGUI.class.getResource("/img/icon-digital-nao-
verificada.png"));
            Image image = imageIcon.getImage();
            Image novaImg = image.getScaledInstance(75, 63,
java.awt.Image.SCALE_SMOOTH);
            imageIcon = new ImageIcon(novaImg);
            JOptionPane.showMessageDialog(janelaDono.getJDialog(),"As
impressões digitais não são compatíveis.", "Falha na verificação",
JOptionPane.ERROR_MESSAGE, imageIcon);}}}
    else{trataErrosExcecaoEscaner.erro(ffv.getEngineStatus());}}
protected void carregaListaUsuarios(){
    listaUsuarios = new HashMap<>();
    File arquivoDB = new File(ScannerNffv.getBancoDeDados() + ".fdb");
    if(arquivoDB.exists()){
        try{
            FileInputStream arquivoIn = new FileInputStream(arquivoDB);
            ObjectInputStream arquivo = new ObjectInputStream(arquivoIn);
            for (Usuario usuario = (Usuario)arquivo.readObject(); usuario !=
null; usuario = (Usuario)arquivo.readObject()){
                listaUsuarios.put(usuario.getNomeUsuario(), usuario);
            }
            arquivo.close();
        }catch (EOFException eof){}
        catch (Exception e) {e.printStackTrace();}}}
}

```

Fonte: Elaborado pelos autores

### Programa 3 – Classe ScannerNffv

```

package compartilhada;
public abstract class ScannerNffv {
    static private String bancoDeDados      = "dbUsuarios";
    static private String senhaDB           = "ministerio_da_educacao";
    static private String adminDB           = "adminMMA";
    static private int tipoScanner         = 25;
    static public final int TIMEOUT       = 10000;
    static public final int QT_MAX_USER = 9;
    static private Nffv nffv             = null;
    static private void carrega(){
        ScannerModule[] scanner = new ScannerModule[1];
        scanner[0] = Nffv.getAvailableScannerModules()[tipoScanner];
        try{nffv = new Nffv(bancoDeDados, senhaDB, scanner);}
        catch(Exception e){
            if(e.getMessage().equals("Win32 error has occurred")){
                System.out.println(e.getMessage());
                JOptionPane.showMessageDialog(null, "Já existe uma janela do banco
de dados em execução!", "", JOptionPane.ERROR_MESSAGE);
                System.exit(0);}
        }
    static public Nffv getNffv(){
        if(nffv == null){carrega();} return nffv;
    }
}

```

```

    static public String getBancoDeDados() { return bancoDeDados; }
    static public String getSenhaDB() { return senhaDB; }
    static public String getAdminDB() { return adminDB; }
}

```

Fonte: Elaborado pelos autores

#### Programa 4 – Classe Usuario

```

package compartilhada;
public class Usuario implements Serializable{
    private int id;
    private String nome;
    private String nomeUsuario;
    private int nivelAcesso;
    public Usuario(int id, String nome) {
        this.id = id;
        this.nome = nome;
        this.nivelAcesso = 1;
        this.nomeUsuario = "";
    }
    public void setID(int id) {this.id = id;}
    public int getID() {return id;}
    public void setNome(String nome) {this.nome = nome;}
    public String getNome() {return nome;}
    public void setNomeUsuario(String nomeUsuario) {
        this.nomeUsuario = nomeUsuario;
    }
    public String getNomeUsuario() {return nomeUsuario;}
    public int getNivelAcesso() {return nivelAcesso;}
    public void setNivelAcesso(int nivelAcesso) {
        this.nivelAcesso = nivelAcesso;
    }
    public String toString(){return nome;}
}

```

Fonte: Elaborado pelos autores

#### Programa 5 – Classe LoginAdminController

```

package janelaCadastroUsuarios;
public class LoginAdminController implements ActionListener {
    private Principal janelaPrincipal;
    private JanelaGUI janelaGUI;
    private LoginAdminGUI janelaDono;
    private Nffv ffv;
    public LoginAdminController(LoginAdminGUI janelaDono, Principal
janelaPrincipal, JanelaGUI janelaGUI){
        this.janelaDono = janelaDono;
        this.janelaPrincipal = janelaPrincipal;
        this.janelaGUI = janelaGUI;
        this.ffv = ScannerNffv.getNffv();
    }
    protected void atualizaDadosJanela(){
        if(!ffv.getUsers().isEmpty()){
            janelaDono.btnEscanearDigital.setText("escanear digital");
            janelaDono.lblMsgInfo.setText("É preciso verificar se você é o
Administrador");}
    }
    public void actionPerformed(ActionEvent e) {
        if(e.getSource() == janelaDono.btnEscanearDigital){
            if(ffv.getUsers().isEmpty()){
                janelaGUI.janelaCtrl.criaUsuario(ScannerNffv.getAdminDB());
                if(!ffv.getUsers().isEmpty()){
                    adminLogado();}
                else{verificaDigitalAdmin();}}}
    }
}

```

```

private void adminLogado(){
    janelaPrincipal.setAdminLogado(true);
    janelaDono.getJDialog().dispose();
}
private void verificaDigitalAdmin(){
    Usuario adminSelecionado =
(Usuario)janelaGUI.janelaCtrl.listaUsuarios.getElementAt(0);
    NffvUser adminUsuario =
ffv.getUserByID(((Usuario)adminSelecionado).getID());
    int usuarioValidado =
janelaGUI.janelaCtrl.confirmaUsuario(adminUsuario);
    if(usuarioValidado == 1){ adminLogado(); }
    else if(usuarioValidado == 0){
        ImageIcon imageIcon = new
ImageIcon(SobreGUI.class.getResource("/img/icon-digital-nao-
verificada.png"));
        Image image = imageIcon.getImage();
        Image novaImg = image.getScaledInstance(75, 63,
java.awt.Image.SCALE_SMOOTH);
        imageIcon = new ImageIcon(novaImg);
        JOptionPane.showMessageDialog(janelaDono.getJDialog(),"As impressões
digitais não são compatíveis.", "Falha na verificação",
JOptionPane.ERROR_MESSAGE, imageIcon);}}
}

```

Fonte: Elaborado pelos autores

#### Programa 6 – Classe Principal do pacote janelaAcessoDB

```

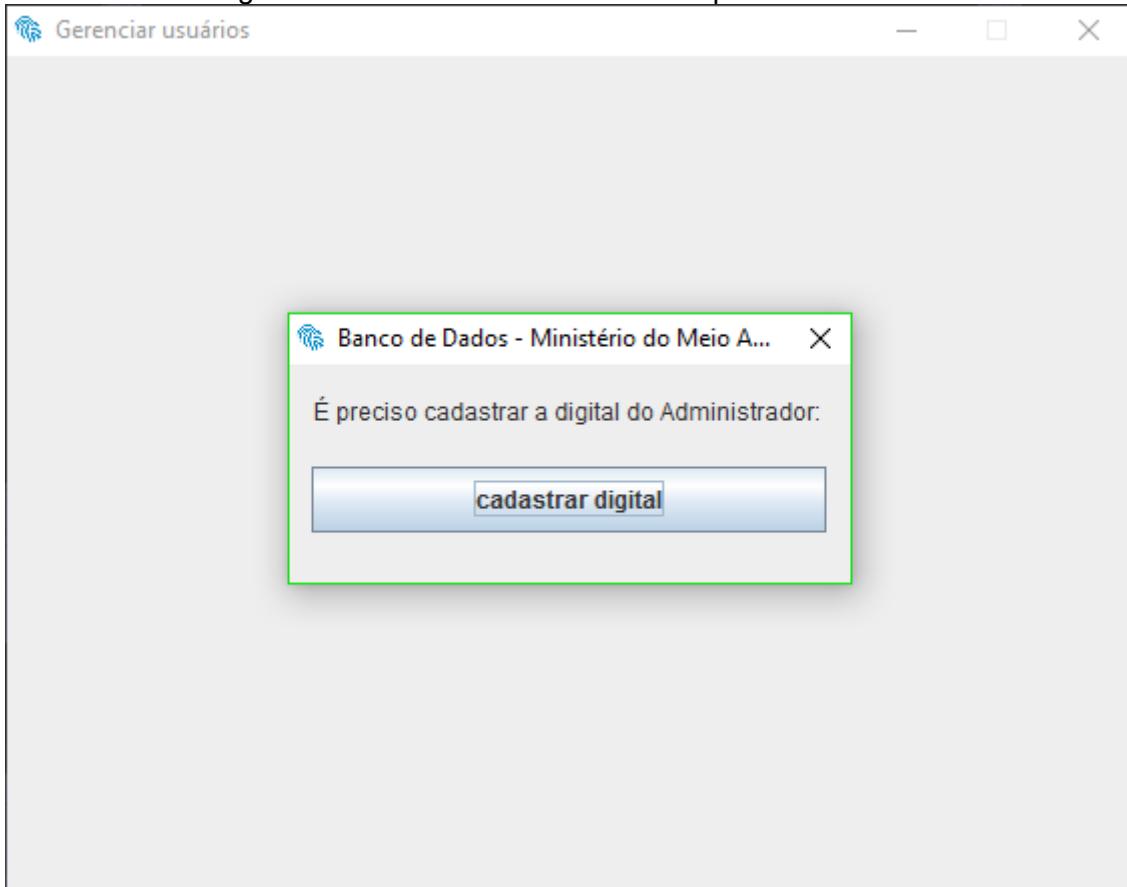
package janelaAcessoDB;
public class Principal {
    private Nffv ffv;
    private Usuario usuarioLogado;
    public static void main(String[] args) {
        new Principal();
    }
    public Principal() {
        ffv = ScannerNffv.getNffv();
        JanelaGUI janelaDB = new JanelaGUI();
        janelaDB.setVisible(true);
        Principal princ = this;
        LoginDBGUI janelaLogin = new LoginDBGUI(princ, janelaDB);
        janelaLogin.getJDialog().setVisible(true);
        while(janelaLogin.getJDialog().isShowing()){}
        if(usuarioLogado != null){
            janelaDB.atualizaDadosUser(usuarioLogado);
            janelaDB.exibe(true);
        }
        else{
            janelaDB.dispose();
        }
    }
    protected Usuario getUsuarioLogado() {
        return usuarioLogado;
    }
    protected void setUsuarioLogado(Usuario usuarioLogado) {
        this.usuarioLogado = usuarioLogado;
    }
}

```

Fonte: Elaborado pelos autores

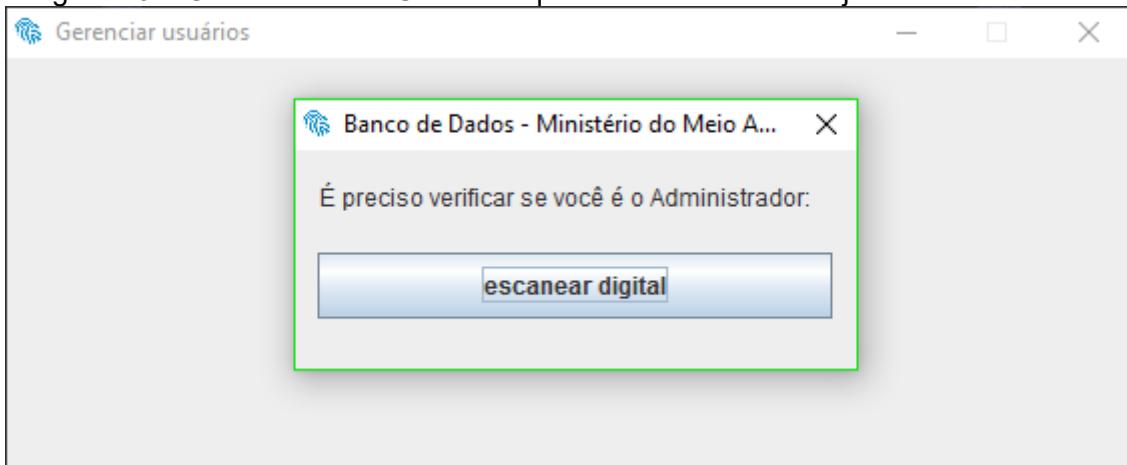
## 7 APRESENTAÇÃO DO SISTEMA EM FUNCIONAMENTO EM UM COMPUTADOR

Figura 15 – Gerenciador de Usuários: primeiro acesso



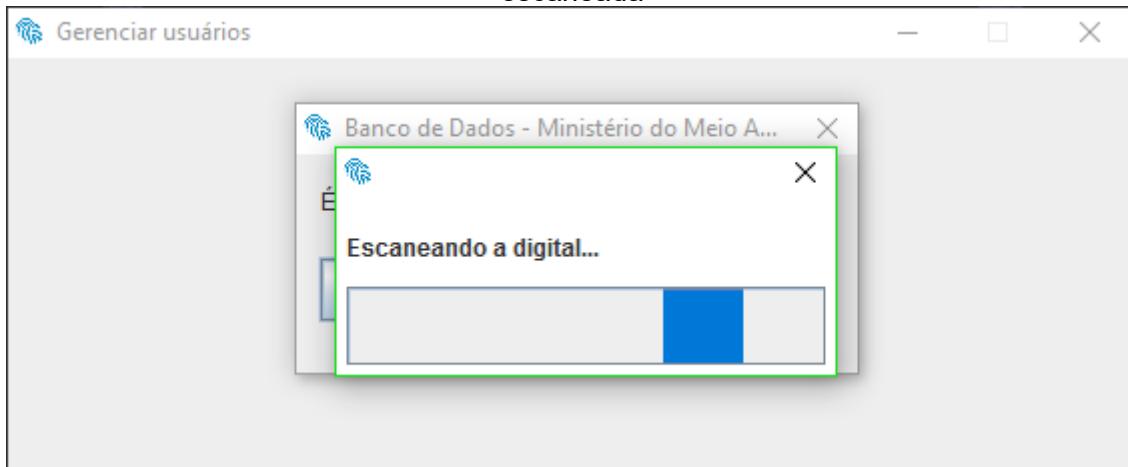
Fonte: Elaborada pelos autores

Figura 16 – Gerenciador de Usuários: quando o administrador já estiver cadastrado



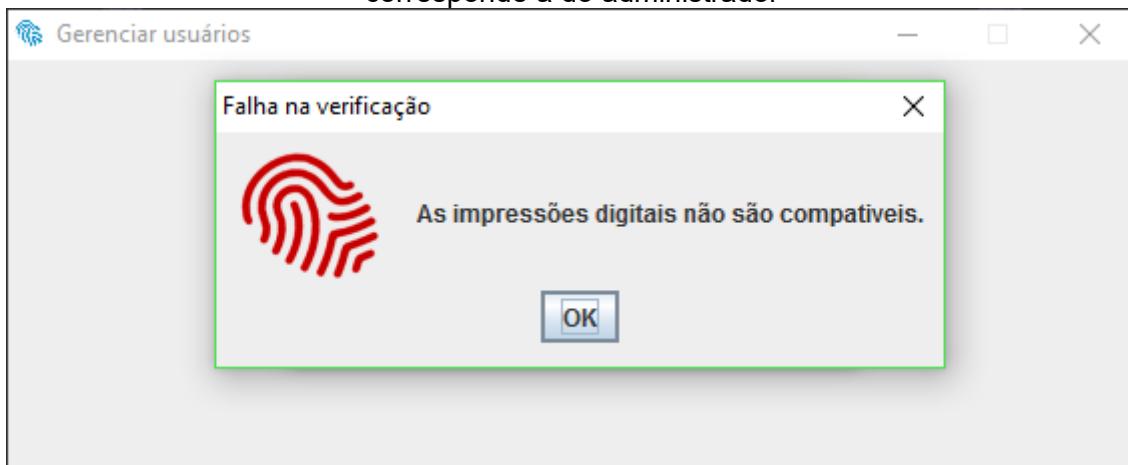
Fonte: Elaborada pelos autores

Figura 17 – Gerenciador de Usuários: mensagem informando que a digital está sendo escaneada



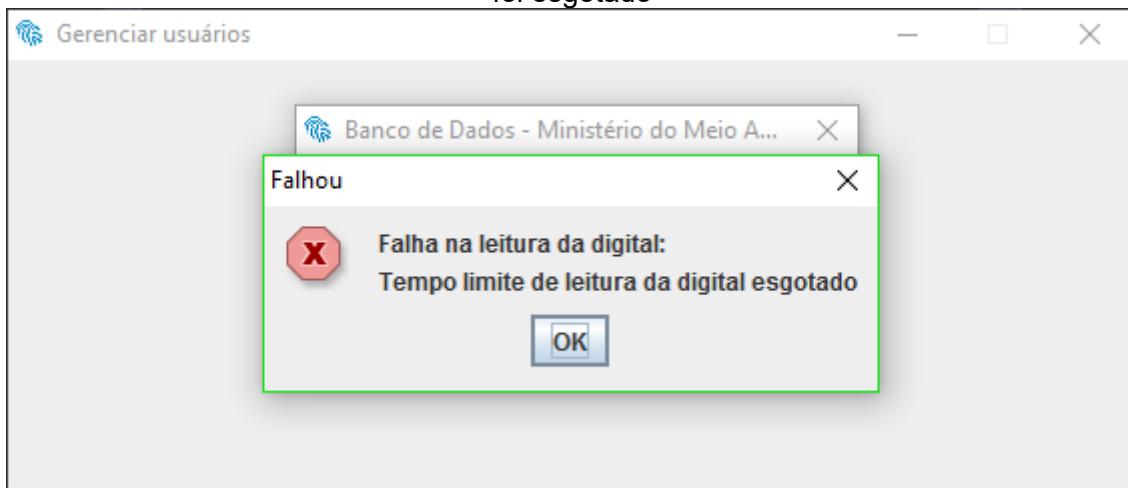
Fonte: Elaborada pelos autores

Figura 18 – Gerenciador de Usuários: tela informando que a digital escaneada não corresponde a do administrador



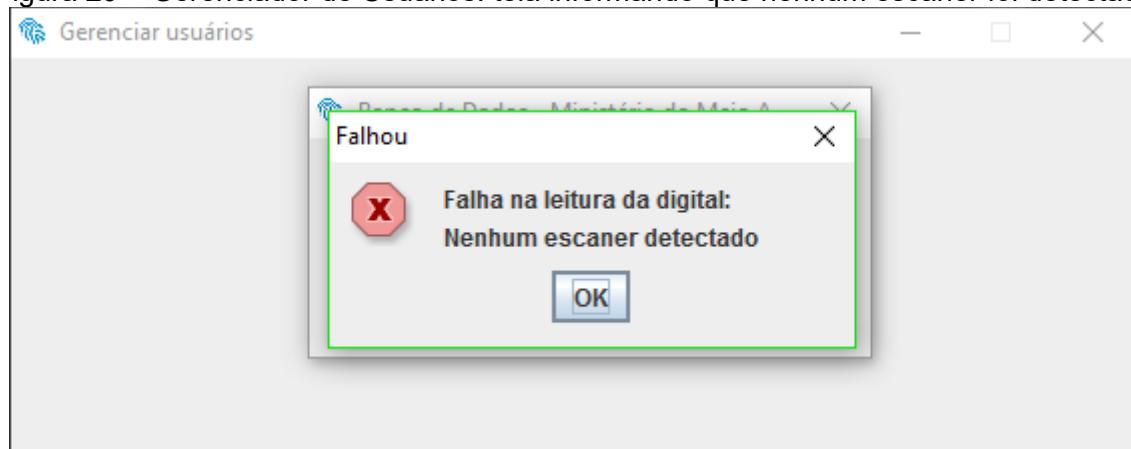
Fonte: Elaborada pelos autores

Figura 19 – Gerenciador de Usuários: tela informando que o tempo limite de leitura da digital foi esgotado



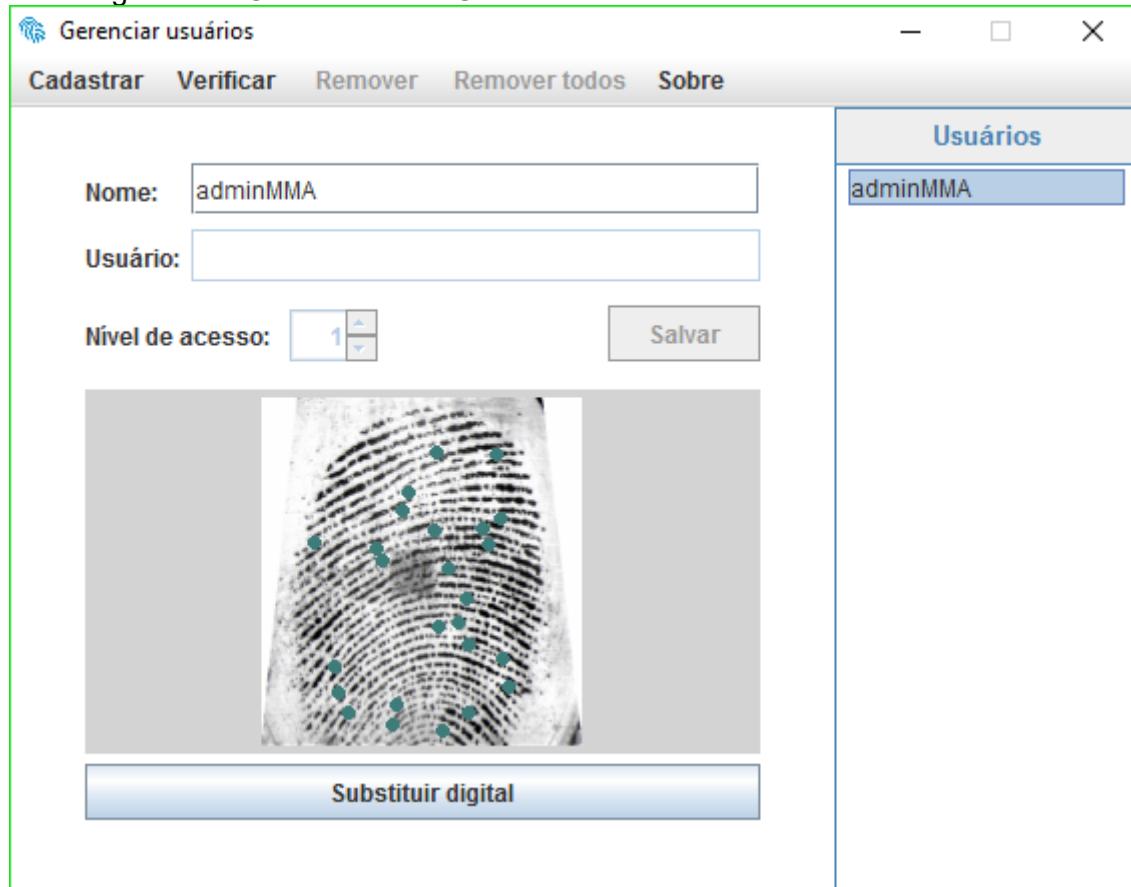
Fonte: Elaborada pelos autores

Figura 20 – Gerenciador de Usuários: tela informando que nenhum escaner foi detectado



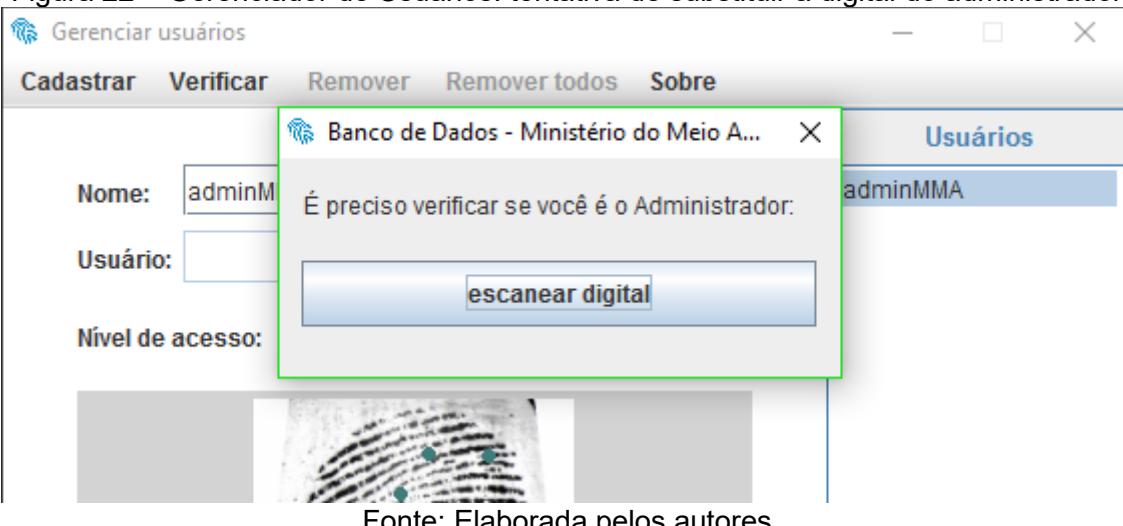
Fonte: Elaborada pelos autores

Figura 21 – Gerenciador de Usuários: somente o administrador cadastrado



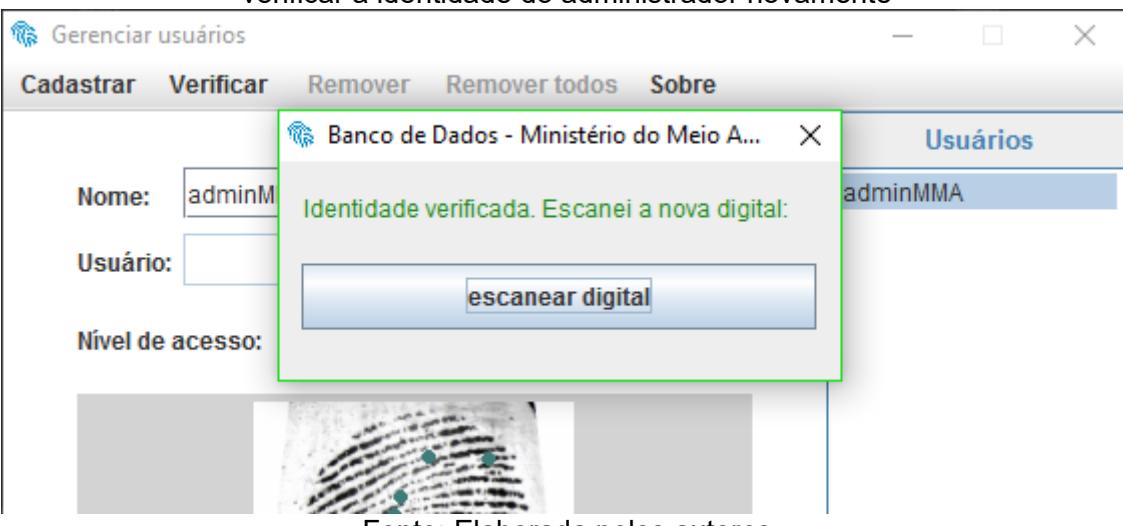
Fonte: Elaborada pelos autores

Figura 22 – Gerenciador de Usuários: tentativa de substituir a digital do administrador



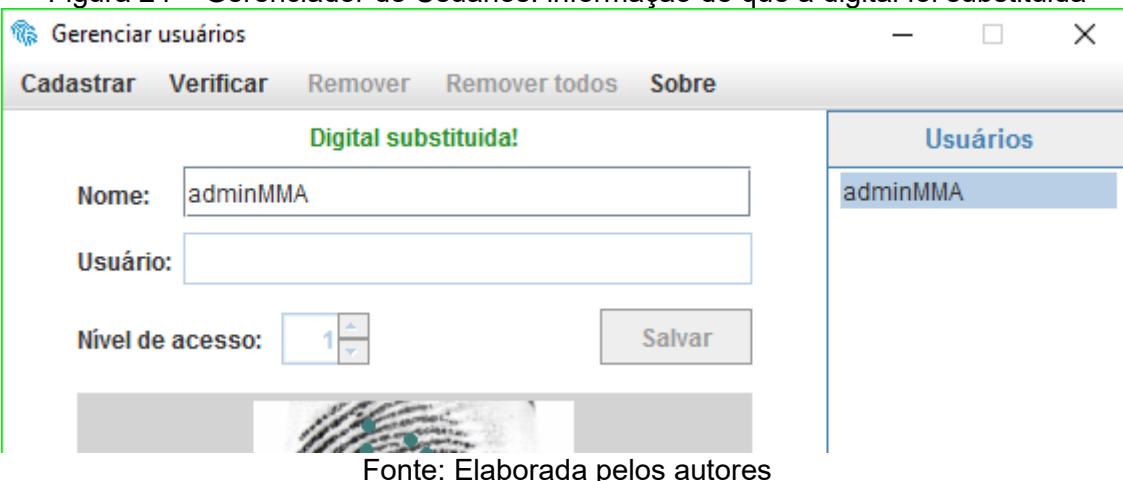
Fonte: Elaborada pelos autores

Figura 23 – Gerenciador de Usuários: administrador verificado. Após 1m será necessário verificar a identidade do administrador novamente



Fonte: Elaborada pelos autores

Figura 24 – Gerenciador de Usuários: informação de que a digital foi substituída



Fonte: Elaborada pelos autores

Figura 25 – Gerenciador de Usuários: cadastrar um usuário

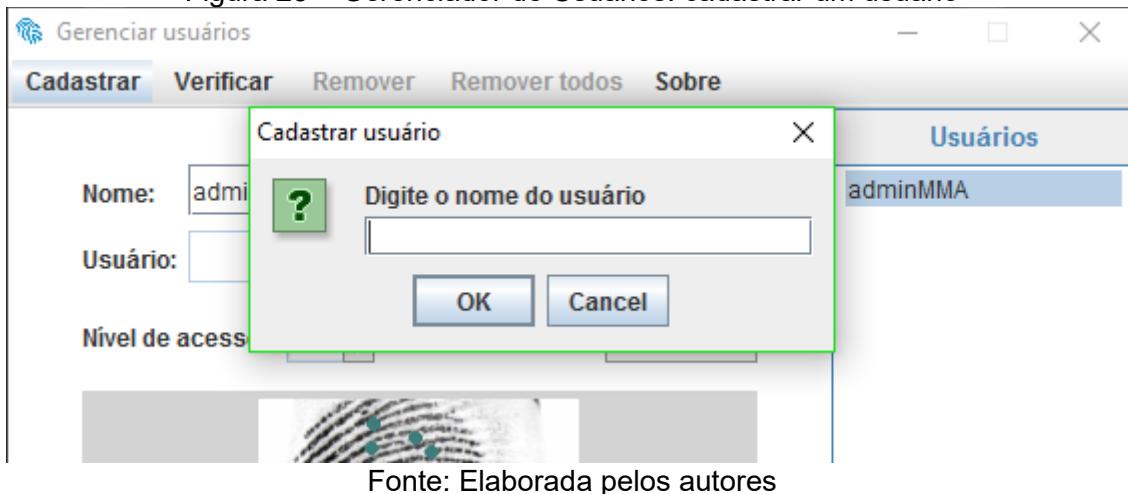


Figura 26 – Gerenciador de Usuários: escaneando a digital do novo usuário

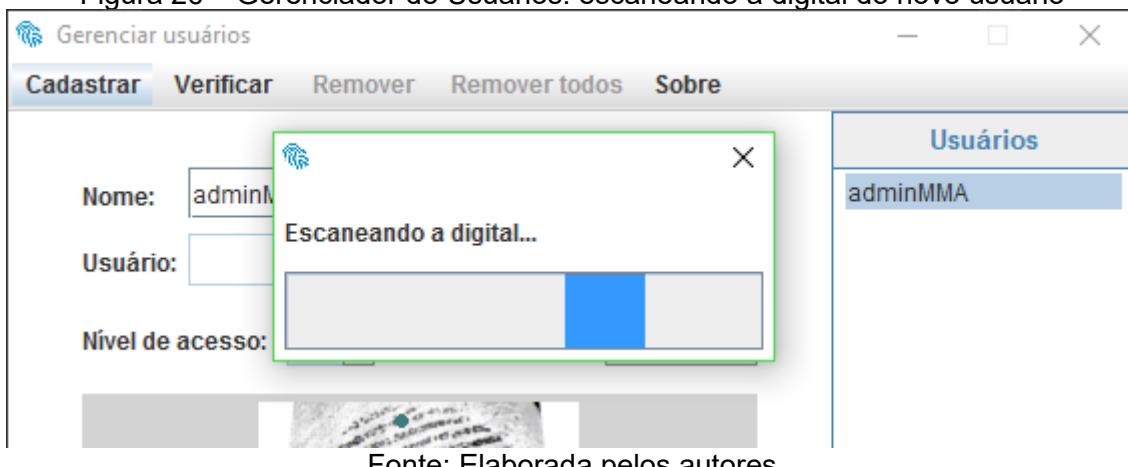
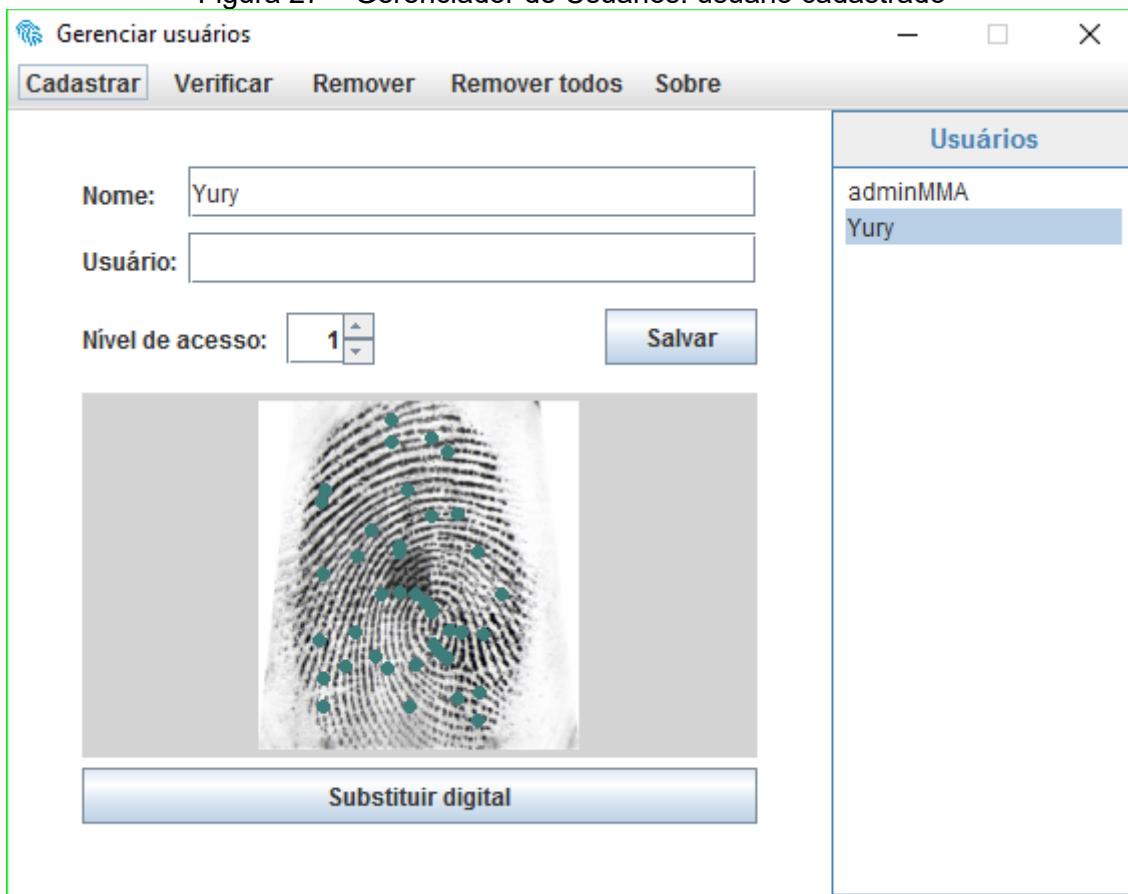
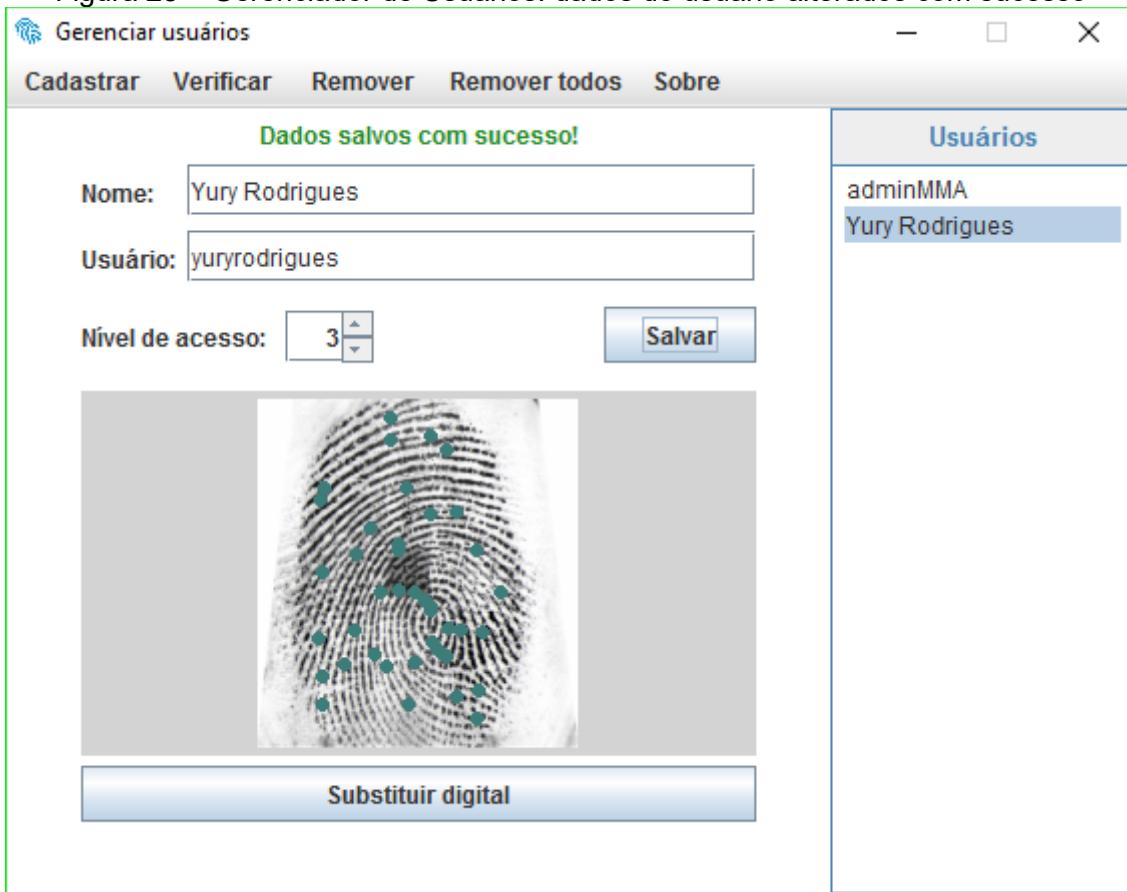


Figura 27 – Gerenciador de Usuários: usuário cadastrado



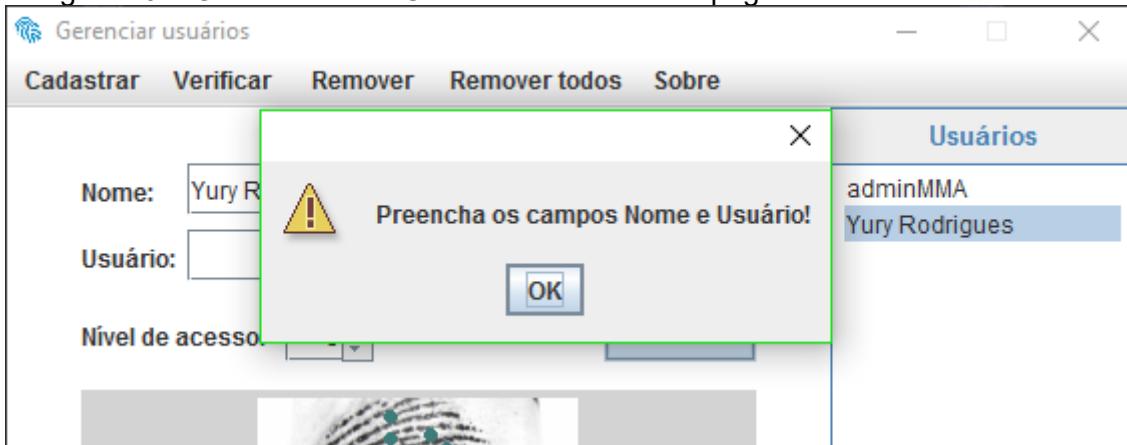
Fonte: Elaborada pelos autores

Figura 28 – Gerenciador de Usuários: dados do usuário alterados com sucesso



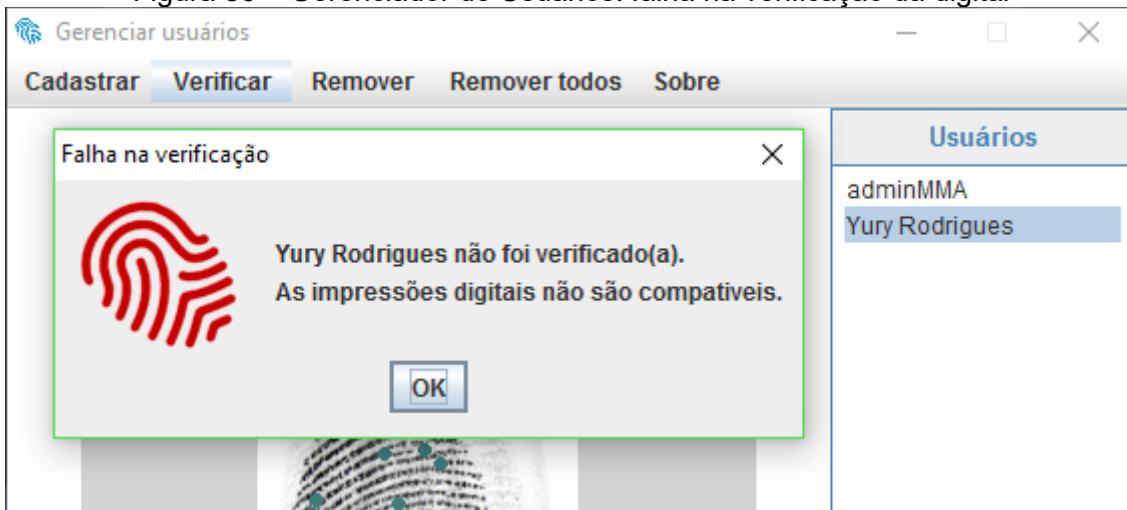
Fonte: Elaborada pelos autores

Figura 29 – Gerenciador de Usuários: tentativa de apagar os dados de um usuário



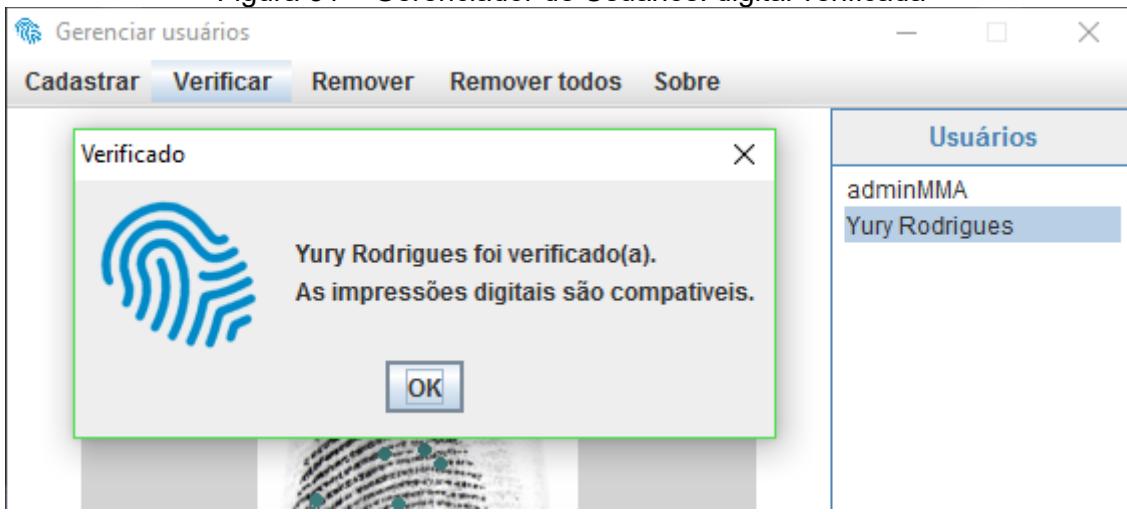
Fonte: Elaborada pelos autores

Figura 30 – Gerenciador de Usuários: falha na verificação da digital



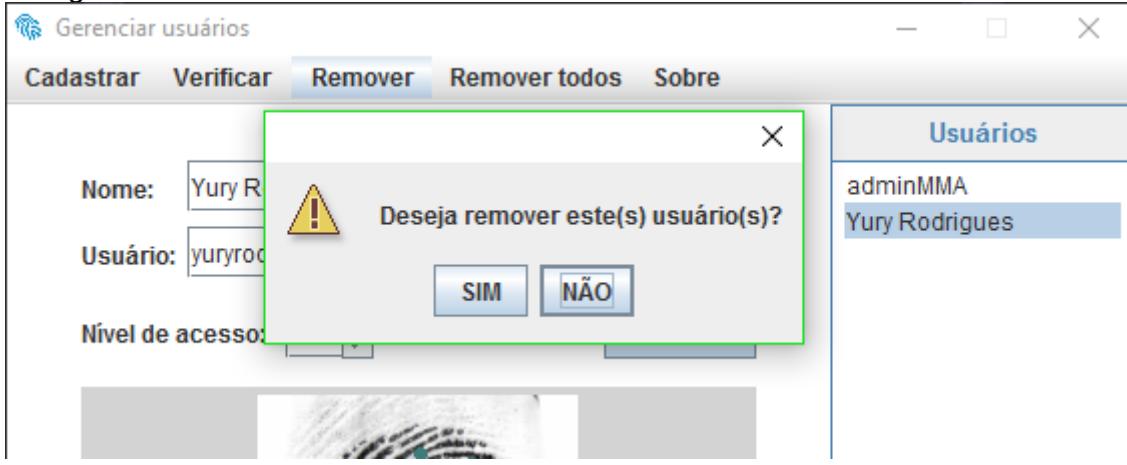
Fonte: Elaborada pelos autores

Figura 31 – Gerenciador de Usuários: digital verificada



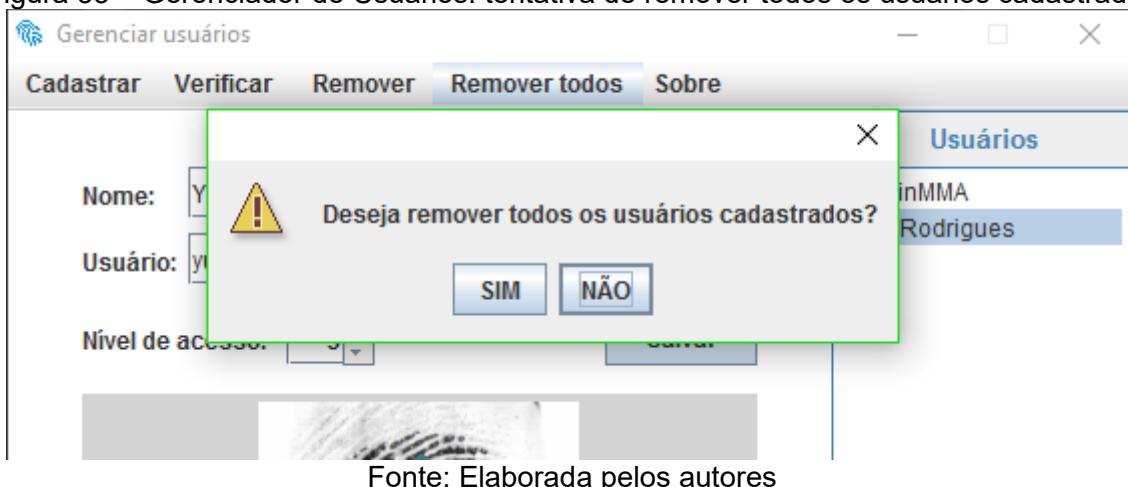
Fonte: Elaborada pelos autores

Figura 32 – Gerenciador de Usuários: tentativa de remover usuários cadastrados



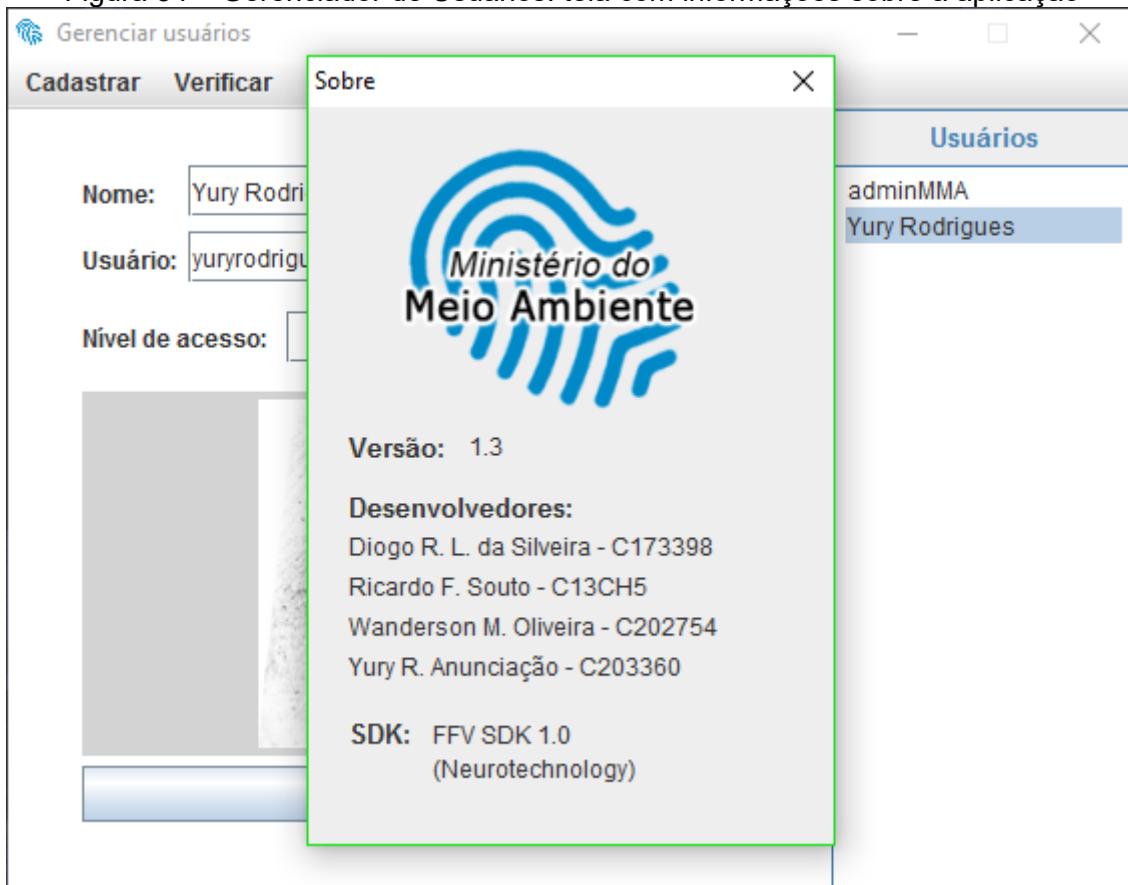
Fonte: Elaborada pelos autores

Figura 33 – Gerenciador de Usuários: tentativa de remover todos os usuários cadastrados



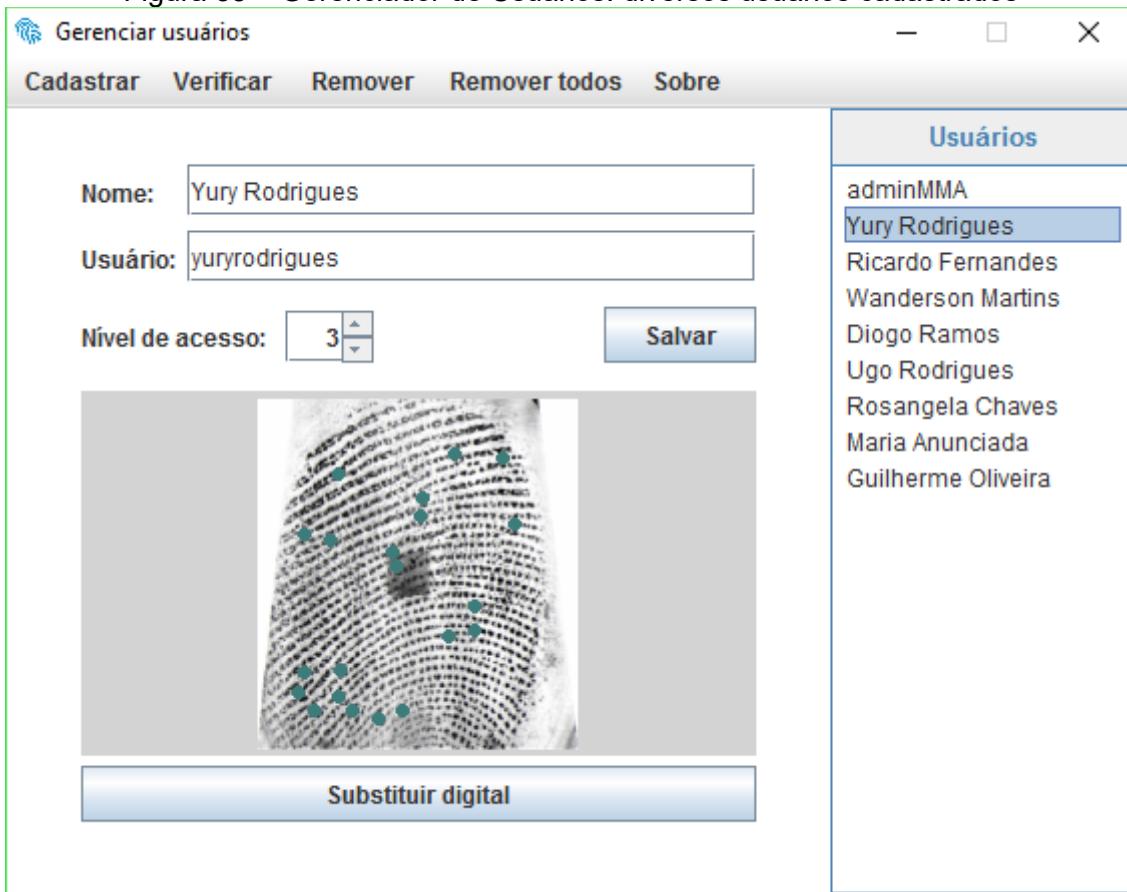
Fonte: Elaborada pelos autores

Figura 34 – Gerenciador de Usuários: tela com informações sobre a aplicação



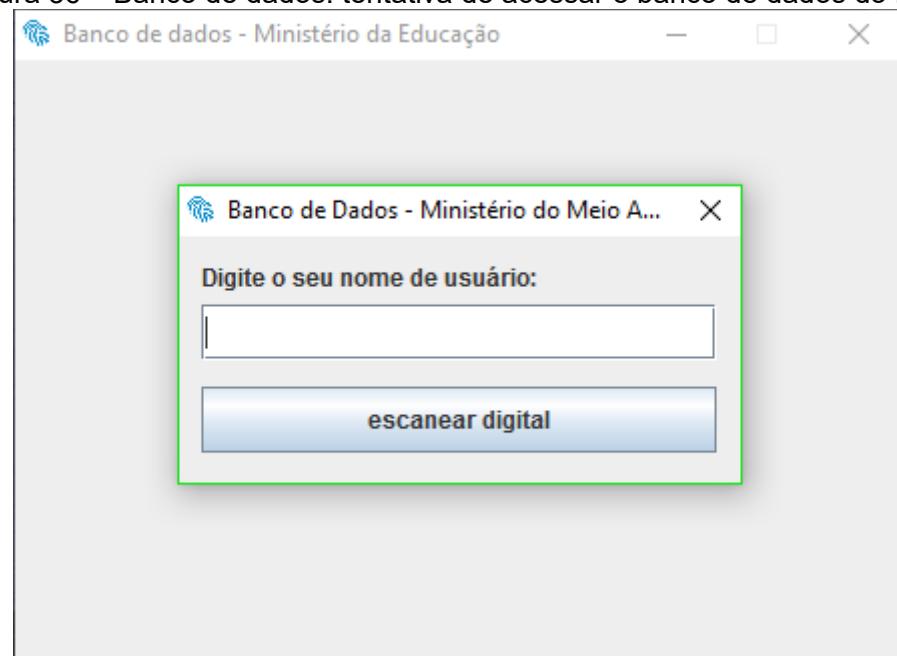
Fonte: Elaborada pelos autores

Figura 35 – Gerenciador de Usuários: diversos usuários cadastrados



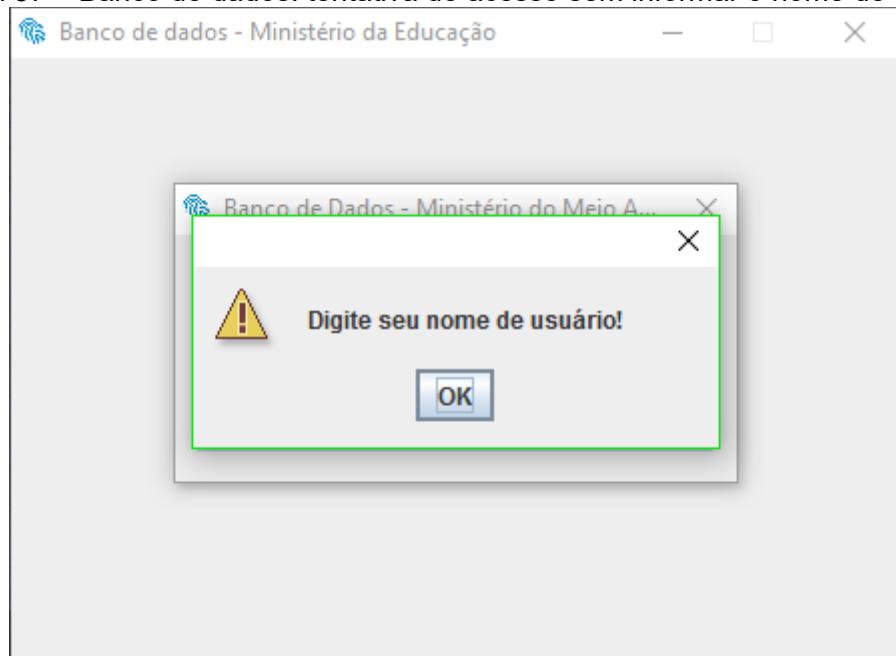
Fonte: Elaborada pelos autores

Figura 36 – Banco de dados: tentativa de acessar o banco de dados do MMA



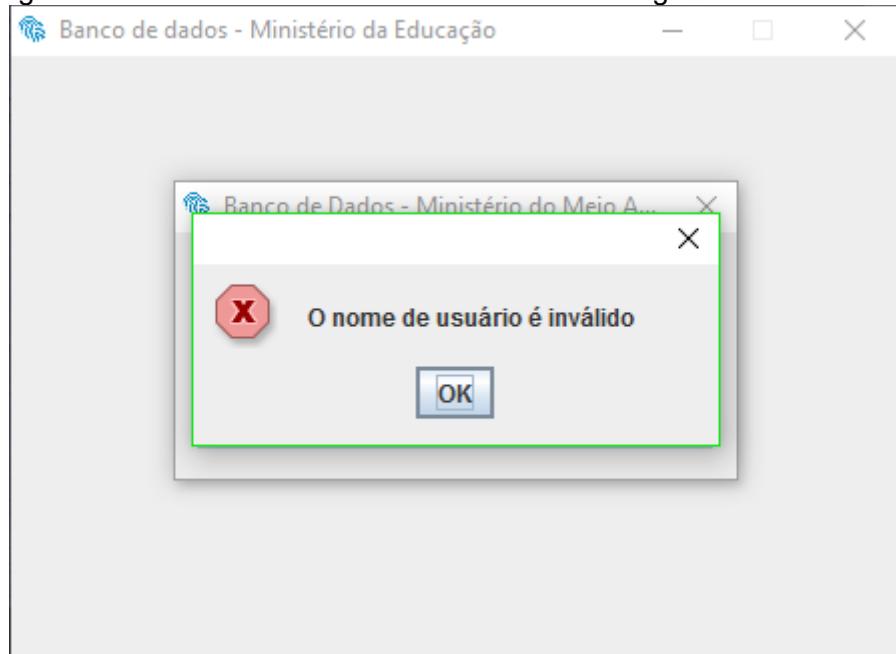
Fonte: Elaborada pelos autores

Figura 37 – Banco de dados: tentativa de acesso sem informar o nome de usuário



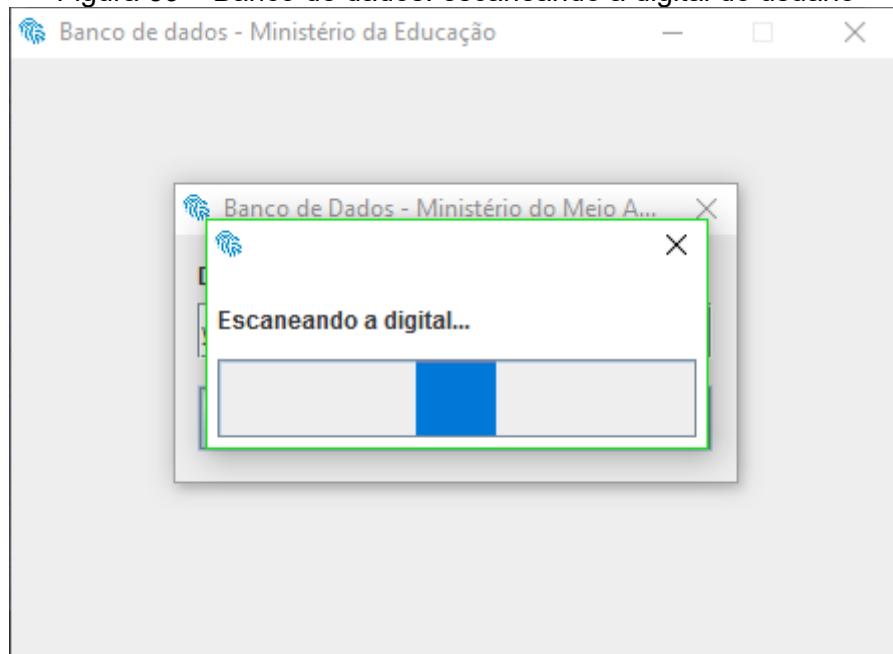
Fonte: Elaborada pelos autores

Figura 38 – Banco de dados: o nome de usuário digitado está incorreto



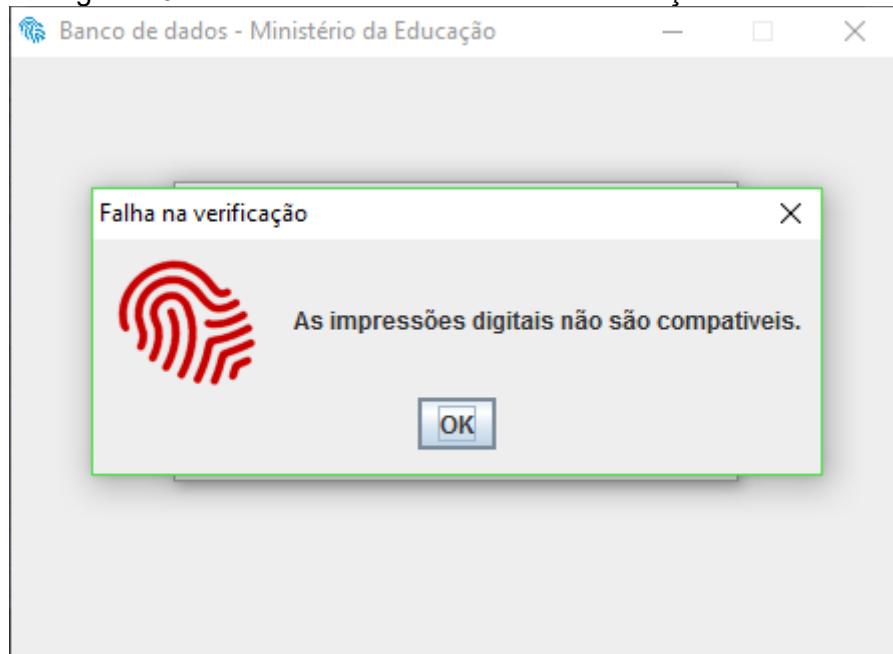
Fonte: Elaborada pelos autores

Figura 39 – Banco de dados: escaneando a digital do usuário



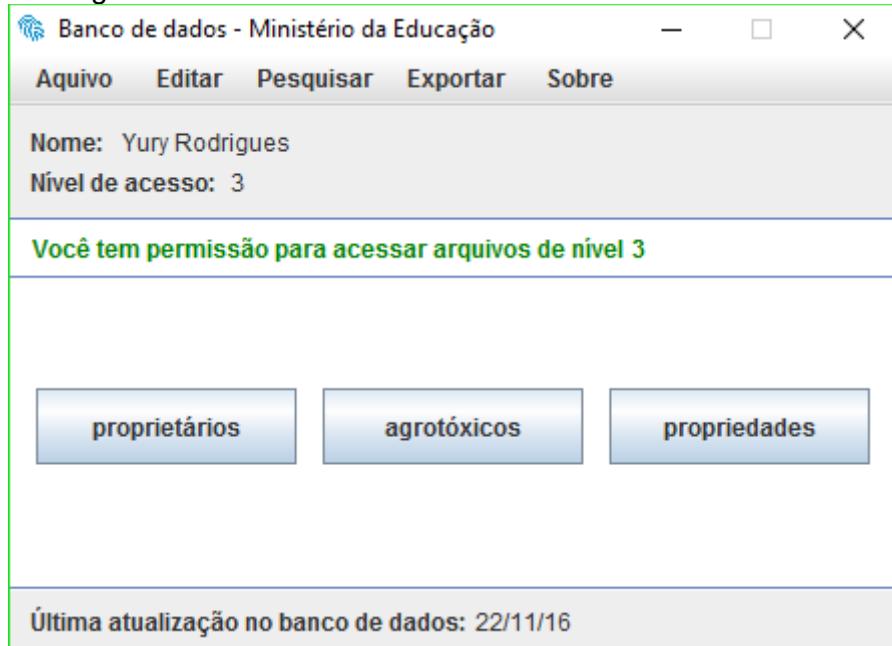
Fonte: Elaborada pelos autores

Figura 40 – Banco de dados: falha na autenticação do usuário



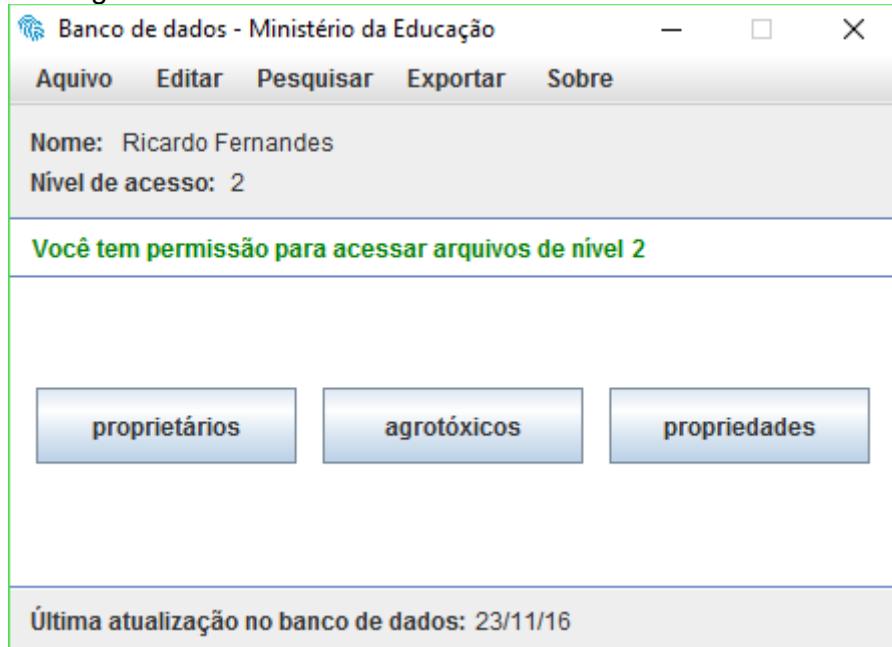
Fonte: Elaborada pelos autores

Figura 41 – Banco de dados: usuário de nível 3 autenticado



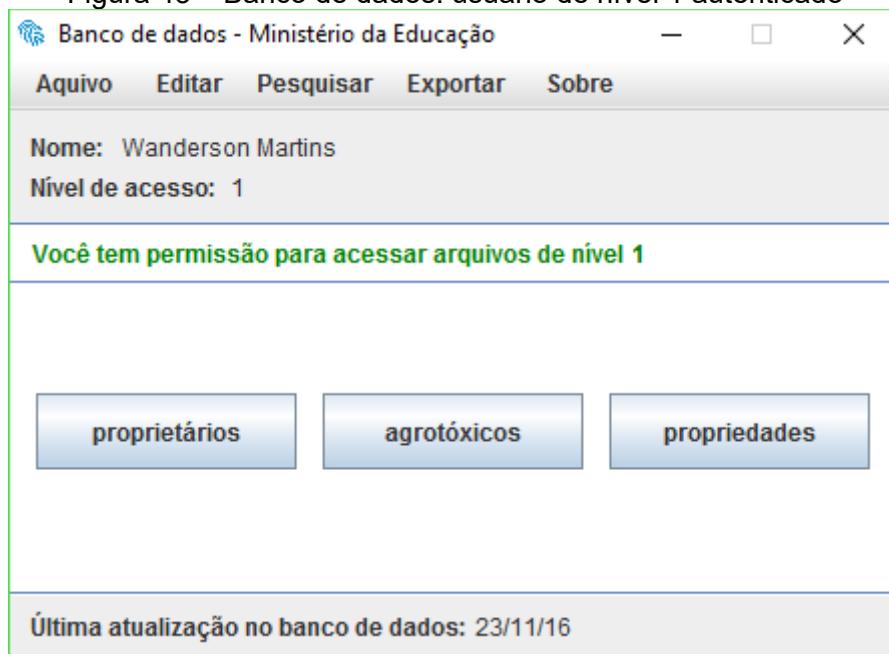
Fonte: Elaborada pelos autores

Figura 42 – Banco de dados: usuário de nível 2 autenticado



Fonte: Elaborada pelos autores

Figura 43 – Banco de dados: usuário de nível 1 autenticado



Fonte: Elaborada pelos autores

## REFERÊNCIAS

**AGÊNCIA BRASIL. Polícia Federal inaugura amanhã moderno sistema de identificação digital.** Disponível em: <<http://memoria.ebc.com.br/agenciabrasil/noticia/2004-08-02/policia-federal-inaugura-amanh%C3%A1-moderno-sistema-de-identificacao-digital>>. Acesso em: 20 nov. 2016.

**ARAÚJO, Paulo Gabriel Ribacionka Góes de.** **Sistema de controle de acesso via smart card com autenticação biométrica da impressão digital.** 2010. 105 f. Trabalho de Conclusão de Curso (Graduação em Engenharia de Computação) – Centro Universitário de Brasília(UniCEUB), Brasília, 2010. Disponível em: <<http://www.repository.uniceub.br/bitstream/123456789/3382/3/20516507.pdf>>. Acesso em: 20 nov. 2016.

**BIER, Carlos Eduardo.** **Bioengine SDK: Identificação Biométrica 1:N.** Disponível em: <<https://www.profissionaisti.com.br/2011/06/bioengine-sdk-identificacao-biometrica-1n/>>. Acesso em: 20 nov. 2016.

**BIOMETRIC SOLUTIONS.** **Face Recognition.** Disponível em: <<http://www.biometric-solutions.com/face-recognition.html>>. Acesso em: 20 nov. 2016.

**BIOMETRIC SOLUTIONS.** **Fingerprint Recognition.** Disponível em: <<http://www.biometric-solutions.com/fingerprint-recognition.html>>. Acesso em: 20 nov. 2016.

**BIOMETRIC SOLUTIONS.** **Iris Recognition.** Disponível em: <<http://www.biometric-solutions.com/iris-recognition.html>>. Acesso em: 20 nov. 2016.

**BIOMETRIC SOLUTIONS.** **Keystroke Dynamic.** Disponível em: <<http://www.biometric-solutions.com/keystroke-dynamics.html>>. Acesso em: 20 nov. 2016.

**BIOMETRIC SOLUTIONS.** **Speaker Recognition.** Disponível em: <<http://www.biometric-solutions.com/speaker-recognition.html>>. Acesso em: 20 nov. 2016.

**BURSZTYN, Victor Soares.** **Biométria: Análise de Assinaturas.** Disponível em: <[http://www.gta.ufrj.br/grad/08\\_1/assinat/](http://www.gta.ufrj.br/grad/08_1/assinat/)>. Acesso em: 20 nov. 2016.

**FARIA, Alessandro de Oliveira.** **Biometria: Processamento de imagens capturadas em leitores de impressão digital.** Disponível em: <<http://www.linhadecodigo.com.br/artigo/1162/biometria-processamento-de-imagens-capturadas-em-leitores-de-impressao-digital.aspx>>. Acesso em: 20 nov. 2016.

**GTA – UFRJ.** **Biometria – Assinatura.** Disponível em: <[http://www.gta.ufrj.br/grad/10\\_1/1a-versao/assinatura/historico.html](http://www.gta.ufrj.br/grad/10_1/1a-versao/assinatura/historico.html)>. Acesso em: 20 nov. 2016.

GTA – UFRJ. **Impressão Digital: Constituição.** Disponível em: <[http://www.gta.ufrj.br/grad/07\\_2/leonardo/Constituio.html](http://www.gta.ufrj.br/grad/07_2/leonardo/Constituio.html)>. Acesso em: 20 nov. 2016.

NEUROTECHNOLOGY. **Free Fingerprint Verification SDK.** Disponível em: <<http://www.neurotechnology.com/free-fingerprint-verification-sdk.html>>. Acesso em: 20 nov. 2016.

ROUSE, Margaret. **Biometrics.** Disponível em: <<http://searchsecurity.techtarget.com/definition/biometrics>>. Acesso em: 20 nov. 2016.

TUTORIALS POINT. **Biometrics: Physiological Modalities.** Disponível em: <[https://www.tutorialspoint.com/biometrics/physiological\\_modalities.htm](https://www.tutorialspoint.com/biometrics/physiological_modalities.htm)>. Acesso em: 20 nov. 2016.

VIVA O LINUX. **Como funcionam os sistemas de biometria: um estudo geral.** Disponível em: <<https://www.vivaolinux.com.br/artigo/Como-funcionam-os-sistemas-de-biometria-um-estudo-geral>>. Acesso em: 20 nov. 2016.

WILSON, Tracy V. **How Biometrics Works.** Disponível em: <<http://science.howstuffworks.com/biometrics.htm>>. Acesso em: 20 nov. 2016.



FICHA DAS ATIVIDADES PRÁTICAS SUPERVISIONADAS - APS

UNIVERSIDADE PAULISTA

NOME: Diego Thomaz Borges da Silveira TURMA: CC6P33 RA: CJ73398  
CURSO: ciéncia da computação CAMPUS: TATUAPÉ SEMESTRE: 6º TURNO: NOTURNO  
CÓDIGO DA ATIVIDADE: 317-X SEMESTRE: 6º ANO GRADE: 2014

DATA DA ATIVIDADE	DESCRÍÇÃO DA ATIVIDADE	TOTAL DE HORAS	ASSINATURA DO ALUNO	HORAS ATRIBUÍDAS (1)	ASSINATURA DO PROFESSOR
20/08/2023	Levantamento bibliográfico Tema Desenvolvimento	3 horas	Diego Ribeiro Diego Ribeiro Diego Ribeiro Diego Ribeiro	3	Diego Ribeiro

(1) Horas atribuídas de acordo com o regulamento das Atividades Práticas Supervisionadas do curso.

TOTAL DE HORAS ATRIBUÍDAS:

## AVALIAÇÃO:

Aprovado ou Reprovado

DATA: \_\_\_\_\_

CARIMBO E ASSINATURA DO COORDENADOR DO CURSO



## FICHA DAS ATIVIDADES PRÁTICAS SUPERVISIONADAS - APS

NOME: Ricardo Fernandes Soárez TURMA: CC6P33 RA: C13CH15

**CURSO:** Ciencia del computo **CAMPUS:** Tuxtla Gómez - **SEMESTRE:** 6<sup>º</sup> **TURNO:** Noturno

CÓDIGO DA ATIVIDADE: 317 SEMESTRE: 6º ANO GRADE: 2014

DATA DA ATIVIDADE	DESCRICAÇÃO DA ATIVIDADE				
TOTAL DE HORAS	ASSINATURA DO ALUNO	HORAS ATRIBUÍDAS (1)	ASSINATURA DO PROFESSOR		
Levantamento bíblico na Fábrica	Temos desenvolvimento	2 horas	2 horas	Assinatura do professor	

(1) Horas atribuídas de acordo com regulamento das Atividades Práticas Supervisionadas do curso.

TOTAL DE HORAS ATRIBUÍDAS:

## AVALIAÇÃO:

Aprovado ou Reprovado

DATA: / /

**CARIMBO E ASSINATURA DO COORDENADOR DO CURSO**



FICHA DAS ATIVIDADES PRÁTICAS SUPERVISIONADAS - APS

UNIVERSIDADE PAULISTA

**NAME:** Wanderson Martins Oliveira

TURMA: CC6P33

RA: C2027S-4

卷之三

CURSO: Língua da Computação CAMPUS: Ativapé SEMESTRE: 02 TURMA: 1014  
CÓDIGO DA ATIVIDADE: 317 X' SEMESTRE: 02 ANO GRADE: 10

CÓDIGO DA ATIVIDADE: 317 SEMESTRE: 6º ANO GRADE: 10

DATA DA ATIVIDADE	DESCRÍÇÃO DA ATIVIDADE		
TOTAL DE HORAS	ASSINATURA DO ALUNO	HORAS ATRIBUÍDAS (1)	ASSINATURA DO PROFESSOR
Levantamento bibliográfico Tema Desenvolvimento		Ass. Ass. Ass.	

(1) Horas atribuídas de acordo com o regulamento das Atividades Práticas Supervisionadas do curso.

### TOTAL DE HORAS ATRIBUIDAS:

AVALIAÇÃO

Aprovado ou Reprovado

CARIMBO E ASSINATURA DO COORDENADOR DO CURSO



UNIVERSIDADE PAULISTA

## FICHA DAS ATIVIDADES PRÁTICAS SUPERVISIONADAS - APS

NOME: Yury Rodrigues Anunciação TURMA: CC6P33 RA: C20336-0  
CURSO: Ciências da Computação CAMPUS: Tatuapé SEMESTRE: 6º TURNO: Matutino  
CÓDIGO DA ATIVIDADE: 317X SEMESTRE: 6º ANO GRADE: 2014

DATA DA ATIVIDADE	DESCRÍÇÃO DA ATIVIDADE	TOTAL DE HORAS	ASSINATURA DO ALUNO	HORAS ATRIBUÍDAS (1)	ASSINATURA DO PROFESSOR
	Levantamento bibliográfico	00			
	Tema	00			
	Desenvolvimento	00			

(1) Horas atribuídas de acordo com o regulamento das Atividades Práticas Supervisionadas do curso.

**TOTAL DE HORAS ATRIBUÍDAS:** \_\_\_\_\_

AVALIAÇÃO:

Aprovado ou Reprovado

DATA: / /

## CARIMBO E ASSINATURA DO COORDENADOR DO CURSO