SISTEMA DE AUTENTICAÇÃO DE DOIS FATORES COM USO DE BIOMETRIA PARA REDUÇÃO DOS RISCOS DE SEGURANÇA EM SOFTWARES

Vinicius Elias Barabas[1], Marcel Campos Inocencio[2]

Resumo: Como forma de prevenção às vulnerabilidades cibernéticas mais comuns como ataques de força bruta e violações de dados, a utilização de sistemas de autenticação de dois fatores se tornou prevalente nos últimos anos. Porém, os principais métodos utilizados para o processo contém problemas particulares de usabilidade e segurança, o que além de causar outras vulnerabilidades, ocasiona a não adesão dos usuários. O presente trabalho tem por objetivo criar um sistema de autenticação de dois fatores com utilização de características biométricas e o desenvolvimento de uma página de acesso fictícia para a realização de testes do sistema de autenticação. A característica biométrica escolhida foi a impressão digital e o funcionamento do sistema foi avaliado por meio de testes de precisão e eficiência do processo de comparação da impressão digital. O sistema desenvolvido atingiu seus objetivos de forma satisfatória, realizando o reconhecimento de tentativas de autenticação legítimas ou inválidas com precisão em testes controlados sem voluntários. Ademais, não há transferência de dados biométricos por meio da internet.

Palavras-chave: Autenticação; biometria; cibersegurança.

[1]Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense (Unesc), viniciuselias79@gmail.com
[2]Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense (Unesc), marcel.inocencio@gmail.com

ABSTRACT: As a way to prevent common cybersecurity vulnerabilities such as brute-force attacks and data breaches, the use of two-factor authentication systems has become prevalent in recent years. However, the main methods used in this process have specific usability and security issues, which not only lead to additional vulnerabilities but also result in users opting not to adopt them. This study aims to create a two-factor authentication system using biometric features, along with the development of a mock login page for system testing purposes. The chosen biometric feature was the fingerprint, and the system's performance was evaluated through accuracy and efficiency tests of the fingerprint comparison process. The developed system successfully achieved its objectives, accurately recognizing legitimate and invalid authentication attempts in controlled tests without volunteers. Furthermore, no biometric data is transmitted over the internet.

Keywords: Authentication; biometrics; cybersecurity

1 INTRODUÇÃO

A autenticação, em termos de interação humano-computador, se define como uma medida de segurança que requisita que a pessoa determine sua identidade ao solicitar acesso à um recurso, com o objetivo de seus responsáveis identificarem de forma confiável os solicitantes do mesmo (Pratama; Firmansyah, 2021). Segundo Gupta et al. (2023), existem três tipos de fatores de autenticação: posse, que se refere à algo que o solicitante possui, como um cartão; conhecimento, que se refere à algo que o solicitante sabe, como uma senha; e inerência, que se refere à algo que apenas o solicitante tem por natureza, como sua íris ou impressão digital.

Apesar do avanço exponencial na criação de novas tecnologias e serviços, o sistema de autenticação de único fator por conhecimento, mais precisamente por senha, se mantém até hoje como o mais utilizado, devido principalmente a simplicidade e usabilidade do mesmo (Koistinen et al., 2021). Entretanto, este tipo de autenticação possui graves falhas de segurança, principalmente devido ao fator humano (Oesch; Ruoti, 2020).

Como forma de defender os usuários das consequências da reutilização de senhas e violações de dados, a indústria tem focado na implementação de sistemas de autenticação de dois fatores, que se define como um sistema de autenticação que requer a introdução de um fator a mais, além da senha, no momento de acesso aos serviços (Golla et al., 2021). Dentre os métodos mais comuns para o segundo fator estão os aplicativos

geradores de *OTP (one-time password)*, uma senha aleatória gerada a determinado intervalo de tempo; o envio de um código de confirmação por SMS ou e-mail do usuário; notificações nos *smartphones*; e chaves eletrônicas geradoras de senhas aleatórias (Reese et al., 2019).

Entretanto, segundo Das, Wang e Camp (2019) apesar da diversidade de opções, os usuários relatam dificuldade em seu uso. Configuração inicial, não compreensão do *design* do sistema de autenticação e compatibilidade limitada com seus dispositivos figuram entre os principais problemas que levam à não adesão dos sistemas de forma geral. No caso dos aplicativos geradores de *OTP*, a exclusão do aplicativo, troca, perda ou roubo dos *smartphones* também são desafios em relação à adesão (Gilsenan et al., 2023).

Dutson et al. (2019) demonstra em sua pesquisa que 69% dos entrevistados afirmam que o aplicativo de autenticação adiciona uma quantidade inconveniente de tempo na experiência de *login*. Ademais, no experimento de Farke et al. (2020), com a implementação de uma chave eletrônica *USB* como alternativa à utilização de senhas, os resultados corroboram o mesmo fenômeno: os usuários preferem a autenticação por senhas pelo tempo adicionado ao *login*.

Aliado às questões de usabilidade, estes métodos têm seus problemas particulares de segurança. Berrios et al. (2023) identificou brechas graves em aplicativos populares geradores de *OTP*, que possibilitam a um mau ator comprometer todo o sistema de autenticação, como o armazenamento local de informações referentes ao usuário em arquivos de texto puro, sem criptografia, incluindo as chaves secretas utilizadas para a geração da senha aleatória, endereços de *e-mail* e informações de localização geográfica.

A biometria como fator de autenticação obteve importância em anos recentes graças aos avanços tecnológicos em artefatos de reconhecimento das características biométricas (íris, impressão digital, voz, etc.). Como resultado, sua implementação em aplicações industriais e comerciais é cada vez mais comum (Mansour et al., 2024). Neste contexto, a utilização de impressões digitais é a variação mais popular devido à sua unicidade: cada pessoa possui um padrão de impressões digitais exclusivo, o que torna o método extremamente confiável e amplamente adotado no mercado (Pathan et al., 2019).

Dentre os trabalhos existentes sobre biometria como forma de autenticação, o artigo de Köhler et al. (2021) estudou e classificou em uma

arquitetura de princípios de segurança seis exemplos comerciais de Sistemas de Autenticação Biométrica Remota, que consistem essencialmente em autenticação para serviços Web. Os resultados mostram que apesar de cinco dos seis produtos oferecerem baixo risco de exposição dos dados biométricos, estes têm de confiar o processo de autenticação ao celular do usuário ou realizar a transferência de dados biométricos ao servidor por meio da internet.

O estudo de Bian et al. (2020) propõe a utilização de um identificador e uma chave secreta criada pelo próprio usuário em conjunto com sua impressão digital e um circuito de hardware que gera uma chave teoricamente impossível de ser clonada. O resultado foi positivo, gerando um modelo de autenticação válido e robusto contra ataques cibernéticos conhecidos.

Segoro e Putro (2020) demonstram um modelo de autenticação de dois fatores em um aplicativo de conversas. *QR Codes* e *OTP* são utilizados no momento de registro e login. As impressões digitais são utilizadas no momento de envio de mensagens. Os resultados foram satisfatórios e protegem a aplicação contra o roubo de contas, porém, segundo os autores, mais testes devem ser realizados.

Shah e Kanhere (2019) propõe um sistema de autenticação de dois fatores pela atividade cardiovascular dos indivíduos. O funcionamento do sistema consiste em captar a perturbação causada pela respiração e atividade cardíaca no sinal Wi-Fi. O indivíduo precisa permanecer parado por alguns segundos. O dispositivo atingiu uma precisão média de 84% a 65% em grupos de 2 a 5 pessoas, respectivamente.

O artigo de Kamiński et al. (2023) apresenta uma solução de autenticação de dois fatores utilizando um sistema de captura de voz do usuário. O sistema captura uma gravação de 25 segundos da voz do indivíduo para armazenamento, e nos acessos subsequentes requisita uma gravação de 5 segundos para comparação. O sistema atingiu precisão de 91,82% no reconhecimento de usuários em testes realizados com 55 pessoas.

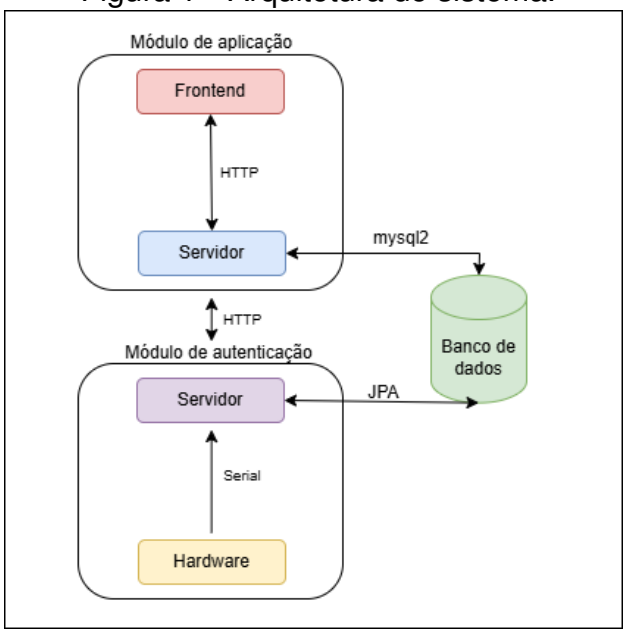
Desse modo, o trabalho desenvolvido no presente artigo resultou em um protótipo de autenticação de dois fatores com a utilização de um leitor de impressões digitais aliado à um *software* para o armazenenamento das informações de autenticação dos usuários de forma criptografada. Além disso, o protótipo não realiza a comunicação de dados biométricos por meio da internet, com o objetivo de reforçar a segurança de acesso à sistemas corporativos que gerenciam dados sensíveis. O sistema com-

pleto visa oferecer maior usabilidade e segurança do que soluções mais populares de autenticação de dois fatores como os aplicativos de celular e seu funcionamento é demonstrado por meio de uma página de acesso à uma aplicação corporativa fictícia. Os objetivos específicos da pesquisa consistem em: compreender o conceito de autenticação de dois fatores, desenvolver um protótipo físico de um sistema de autenticação de dois fatores com biometria, desenvolver um *software* para comunicação e armazenamento criptografado dos dados do protótipo, desenvolver sistema simulado para demonstração do funcionamento do protótipo e articular testes de validação dos dados do protótipo.

2 MATERIAIS E MÉTODOS

A presente pesquisa é aplicada e com base tecnológica. Desenvolveu-se um sistema de autenticação de dois fatores utilizando impressões digitais e também uma aplicação fictícia para validação do fluxo de acesso. O sistema completo constitui-se em dois módulos: de aplicação e de autenticação. Ambos têm acesso ao banco de dados MySQL por meio dos recursos correspondentes: para o módulo aplicação, a biblioteca mysql2; e para o módulo de autenticação, a API *(Application Programming Interface)* de persistência de dados JPA *(Java Persistence API)*. Esta arquitetura é demonstrada na Figura 1.

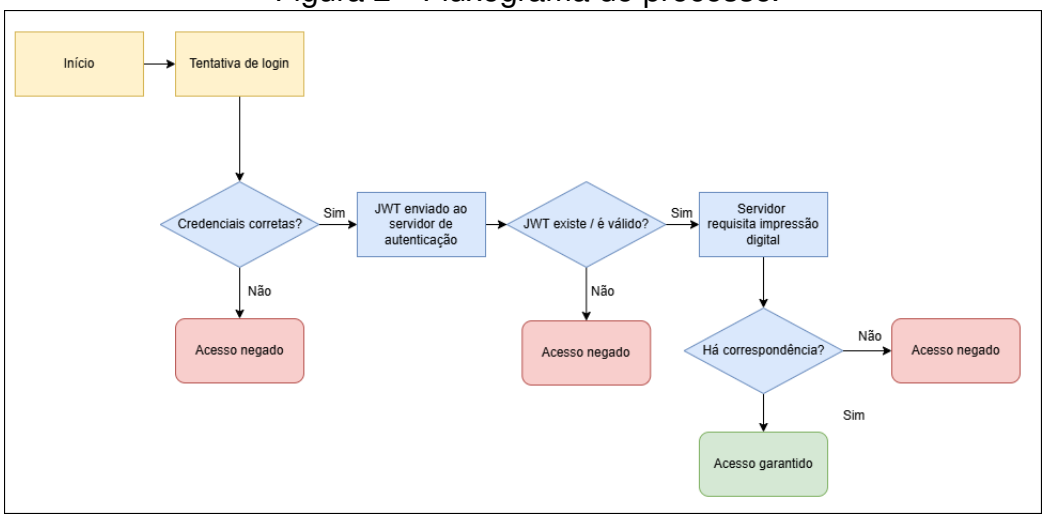
Figura 1 - Arquitetura do sistema.



Fonte: Elaborado pelo autor.

Utilizou-se o módulo de aplicação como base para a implementação do módulo de autenticação. No contexto deste trabalho, ele foi idealizado como uma aplicação fictícia para armazenamento de arquivos dos usuários. O módulo de autenticação constitui-se nos artefatos de *software* e *hardware* na forma de um servidor local Java e um ESP32 com um leitor de impressões digitais ZA620_M5, respectivamente. A comunicação entre os módulos é feita por meio do protocolo HTTP *(Hypertext Transfer Protocol)*. O fluxograma da Figura 2 demonstra o processo que foi implementado no trabalho.

Figura 2 - Fluxograma do processo.

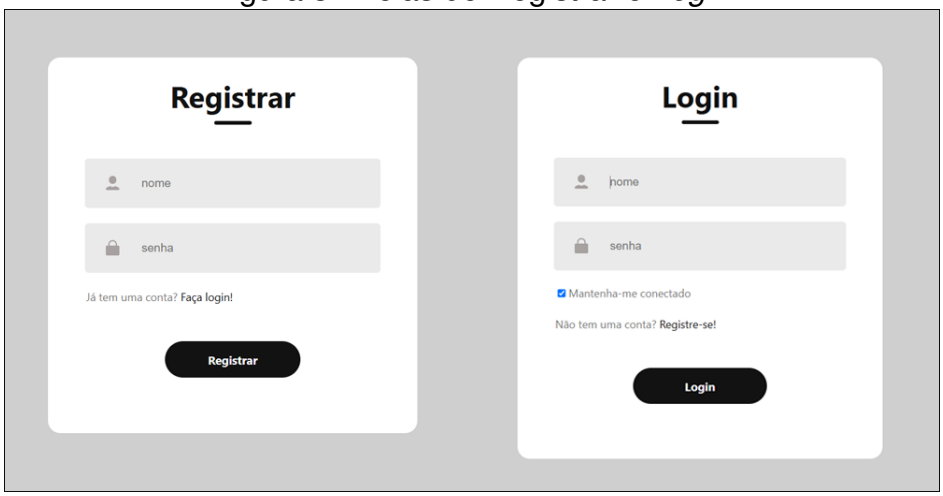


Fonte: Elaborado pelo autor.

2.1 MÓDULO DE APLICAÇÃO

O objetivo do módulo de aplicação é a simulação de um sistema com fluxo de acesso por meio de contas pessoais para a implementação do módulo de autenticação. Desta forma, desenvolveu-se as telas de Registrar e *Login* para a criação e acesso às contas dos usuários. As telas são demonstradas na Figura 3.

Figura 3 - Telas de Registrar e *Login*.



Fonte: Elaborado pelo autor.

Para o desenvolvimento do módulo utilizou-se a linguagem JavaScript, a IDE Visual Studio Code na versão 1.99.3 e o ambiente de execução Node.js na versão 22.14.0. Para o artefato Frontend, foram utilizadas bibliotecas que estruturam a interface e facilitam a interação com o Servidor: react (19.0.0) para construção de componentes dinâmicos, axios (1.8.4) para realização de requisições HTTP, react-router-dom (7.5.0) para controle de rotas na aplicação e sweetalert2 (11.17.2) para exibição de alertas customizados.

Para o artefato Servidor, foram utilizadas as bibliotecas: express (4.21.2) para criação do servidor e definição das rotas, cors (2.8.5) para habilitar requisições entre origens diferentes, cookie-parser (1.4.7) para manipulação de cookies, dotenv (16.4.7) para gerenciamento de variáveis de ambiente, jsonwebtoken (9.0.2) para autenticação via tokens JWT, bcrypt (5.1.1) para criptografia de senhas, e mysql2 (3.13.0) para conexão com o banco de dados MySQL. Todas as bibliotecas são gratuitamente disponibilizadas pelo gerenciador de pacotes NPM (*Node Package Manager*).

A ação do botão de registrar submete os dados inseridos pelo usuário ao Servidor por meio de uma requisição HTTP do método POST. No recebimento desta requisição, o Servidor, por meio da biblioteca bcrypt, gera um *salt*, que consiste em uma palavra com caracteres aleatórios, concatena-o com a senha e aplica uma função de *hash*, procedimento que transforma uma entrada de tamanho variável em uma saída de tamanho fixo. O servidor salva apenas este resultado no banco de dados, evitando o armazenamento de senhas em texto puro, o que proporciona maior segurança. Um exemplo de *string* resultante é demonstrado na Figura 4. A

função de hash é executada 2^n vezes, onde n significa o custo definido pelo programador. No exemplo, a função de hash tem custo 10, ou seja, foi executada 1024 vezes antes de seu resultado ser salvo. Quanto maior o custo, mais lento é o tempo de processamento do resultado, porém mais seguro contra ataques de força bruta. Neste trabalho o custo utilizado foi 10.

Figura 4 - Exemplo de resultado da função hash.

```
$2b$10$nOUIs5kJ7naTuTFkBy1veuK0kSxUFXfuaOKdOKf9xYT0KKIGSJwFa
 |  |  |                     |
 |  |  |                     valor-hash = K0kSxUFXfuaOKdOKf9xYT0KKIGSJwFa
 |  |  |
 |  |  salt = nOUIs5kJ7naTuTFkBy1veu
 |  |
 |  custo => 10 = 2^10 rodadas
 |
 identificador do algoritmo de hash => 2b = BCrypt
```

Fonte: Adaptado de bcrypt (2023).

A ação do botão *login* realiza uma requisição POST ao Servidor, que recebe os dados do usuário e, também utilizando a biblioteca bcrypt, submete a senha inserida à função de hash com o *salt* previamente armazenado e realiza a comparação dos dois resultados. Caso a comparação seja bem-sucedida, o Servidor responde de maneira positiva. Caso contrário, o alerta exibido na Figura 5 é emitido ao usuário.

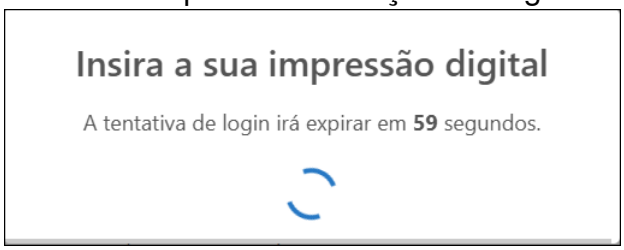
Figura 5 - Alerta emitido ao usuário em caso de senha incorreta.

Fonte: Elaborado pelo autor.

A resposta positiva do Servidor consiste em um JWT *(JSON Web Token)* gerado por meio da biblioteca jsonwebtoken. Ele consiste em um padrão aberto que define uma forma de transmitir informações de maneira segura entre duas partes com um objeto JSON. O *token* tem um tempo de expiração de 60 segundos e é assinado com uma chave privada de 256 bits codificada em formato Base64 e armazenada nas variáveis de ambiente do backend. O Frontend utiliza este *token* para então realizar a autenticação do

segundo fator por meio do alerta emitido ao usuário. Este é demonstrado na Figura 6. O alerta possui um temporizador de 60 segundos e, após expiração, fecha sozinho, requisitando ao usuário nova tentativa de acesso.

Figura 6 - Alerta para autenticação de segundo fator.

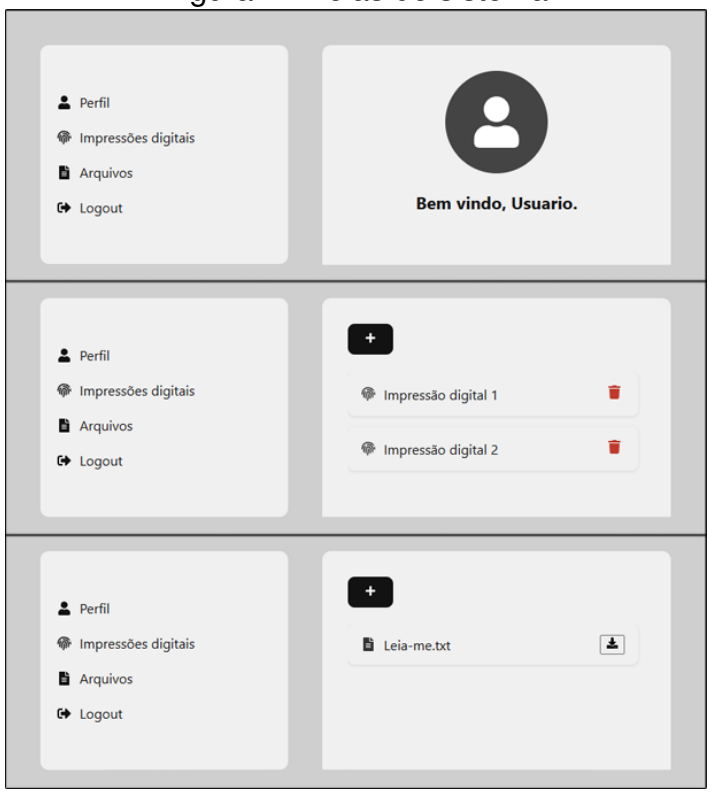


Fonte: Elaborado pelo autor.

O processo de autenticação do segundo fator será iniciado apenas se o *token* informado na requisição for assinado pela chave presente no Servidor e ainda em seu período de validade. Caso ele seja bem-sucedido, o Servidor do módulo de autenticação retorna dois novos JWTs: um *Access Token* e um *Refresh Token*. Estes são utilizados para a autenticação em requisições protegidas do sistema, como a listagem e cadastro de novas impressões digitais. Em particular, o *Refresh Token* é também utilizado para persistir a sessão do usuário por oito horas sem necessidade de nova autenticação, caso o mesmo marque a opção "Mantenha-me conectado" na tela de *login*. O *Access Token* tem validade de 15 minutos e é atualizado automaticamente utilizando o *Refresh Token* caso o mesmo seja válido. Em caso de invalidade de ambos os *tokens*, nenhuma rota protegida é atendida e, caso o usuário tente acessar as telas via URL, o mesmo é encaminhado para novo *login* obrigatório. O funcionamento do algoritmo para a autenticação do segundo fator será explicado na seção posterior.

Após o *login* bem-sucedido, o acesso ao sistema é garantido, sendo o usuário redirecionado para a tela inicial. O sistema possui também a tela de impressões digitais, onde pode ser encontrada a listagem e cadastro de novas impressões digitais, e a tela de arquivos, onde estão listados os arquivos do usuário. As telas do sistema podem ser visualizadas na Figura 7. A navegação entre as telas é realizada pelo menu lateral, que contém ainda a opção de *Logout* para finalizar a sessão do usuário.

Figura 7 - Telas do sistema.



Fonte: Elaborado pelo autor.

2.2 MÓDULO DE AUTENTICAÇÃO

Para a criação do módulo de autenticação, foram desenvolvidos os artefatos de Hardware e Servidor. O Hardware consiste em um microcontrolador ESP32 e o leitor de impressões digitais ZA620_M5. O Servidor é formado por uma API REST desenvolvida na linguagem Java versão 21 com o *framework* Spring Boot versão 3.4.4.

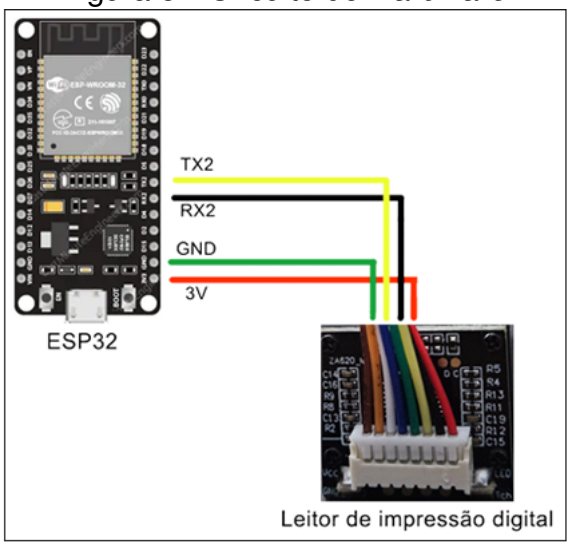
O leitor de impressões digitais escolhido, como outros disponíveis no mercado, possui embutido em seu *firmware* o armazenamento e um algoritmo de comparação de impressões digitais. Entretanto, a utilização dos mesmos implicaria no armazenamento de dados sensíveis em um artefato que pode facilmente ser interceptado por um mau ator. Diante disso, utilizou-se o leitor apenas como receptor da impressão digital, nunca armazenando-a.

2.2.1 Desenvolvimento do Hardware

O ESP32 permite alimentar componentes com tensão de 3V utilizando seu pino 3V3. Além disso, possibilita a comunicação de dados via *Serial* por meio dos pinos RX e TX, devendo estes ser conectados de forma cruzada em relação ao dispositivo externo. Assim, o pino 3V3 foi ligado ao

cabo de alimentação do leitor, o pino RX2 do ESP32 foi conectado ao pino TX do leitor, o pino TX2 do ESP32 foi conectado ao pino RX do leitor, e ambos os pinos GND foram interligados. A alimentação do ESP32 fica a cargo do cabo USB que é conectado ao computador. Este circuito é detalhado na Figura 8.

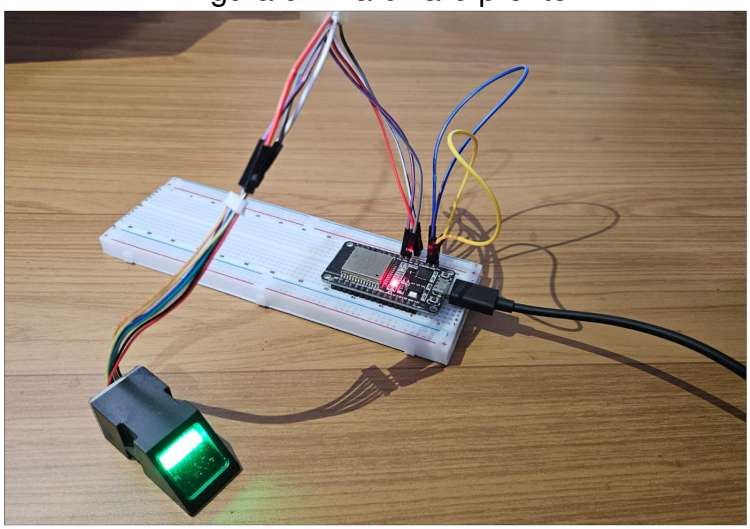
Figura 8 - Circuito do *Hardware*.



Fonte: Elaborado pelo autor.

Como forma de auxiliar a montagem do circuito, utilizou-se uma *protoboard*. O resultado da montagem é demonstrado na Figura 9.

Figura 9 - *Hardware* pronto.



Fonte: Elaborado pelo autor.

O *software* carregado no ESP32 foi desenvolvido com auxílio da biblioteca *open source* FPM, disponível de maneira gratuita no GitHub do

autor Brian Ejike. Ela possui diversas funções utilitárias para trabalhar com os leitores de impressão digital mais populares do mercado IoT *(Internet of Things)*. Dentre estas, o código utilizado foi o de transferência dos bytes da imagem da impressão digital inserida via *Serial* para o computador para posterior utilização pelo Servidor.

2.2.2 Desenvolvimento do Servidor

Para o desenvolvimento do Servidor, foi utilizada a IDE *(Integrated Development Environment)* Intellij IDEA Community Edition em sua versão 2024.3.5. As bibliotecas utilizadas em conjunto com o Spring Boot foram SourceAFIS (3.18.1) para comparação de impressões digitais, jSerialComm (2.6.2) para comunicação com portas seriais do ESP32, dotenv-java (3.2.0) para gerenciamento de variáveis de ambiente e jjwt (0.12.6) para criação e validação de JWTs. Todas as bibliotecas são gratuitamente disponibilizadas pelo repositório de artefatos Maven.

O Servidor disponibiliza *endpoints* para o cadastro de impressões digitais e para a autenticação do segundo fator, de modo que estes necessitam de um *token* de autenticação válido presente na requisição para maior segurança. Ele precisa ser instalado de forma local na máquina do usuário para comunicação com o *Hardware*.

O algoritmo para o cadastro e comparação das impressões digitais foi construído com auxílio da biblioteca *open source* SourceAFIS. Ela disponibiliza a criação de *templates* personalizados a partir de bytes de imagens. Desta maneira, implementou-se um algoritmo que, em comunicação com o *hardware*, captura os bytes transmitidos pelo leitor de impressões digitais pela porta *Serial*, cria uma imagem com 75KB em formato bmp da impressão digital e utiliza os bytes da imagem para montar o *template*. Após isso, o *template* é criptografado com o algoritmo AES utilizando uma chave privada de 256 bits, sendo armazenado no banco de dados para posterior comparação. Após a finalização dos dois processos, a imagem associada à impressão digital é excluída, não existindo armazenamento de dados sensíveis.

2.2.3 Funcionamento do algoritmo de comparação de impressão digital

Para o processo de comparação da impressão digital ao autenticar o segundo fator, a biblioteca SourceAFIS implementa um algoritmo personalizado que consiste na fusão de vários pontos de algoritmos de comparação de impressão digital *open source* diferentes. O algoritmo expõe

estruturas de dados utilizadas internamente durante a comparação, o que o diferencia de outras implementações comerciais e possibilita a realização de projetos abrangentes (Važan, 2023).

A abstração implementada no algoritmo são as minúcias. As minúcias são pontos na imagem da impressão digital com um ângulo de direção correspondente. Estas podem ser definidas também como as terminações ou bifurcações das linhas que formam a impressão digital. Essencialmente, as minúcias são os dados que são salvos nos *templates*. Exemplos de minúcias podem ser visualizados na Figura 10. (Važan, 2023).

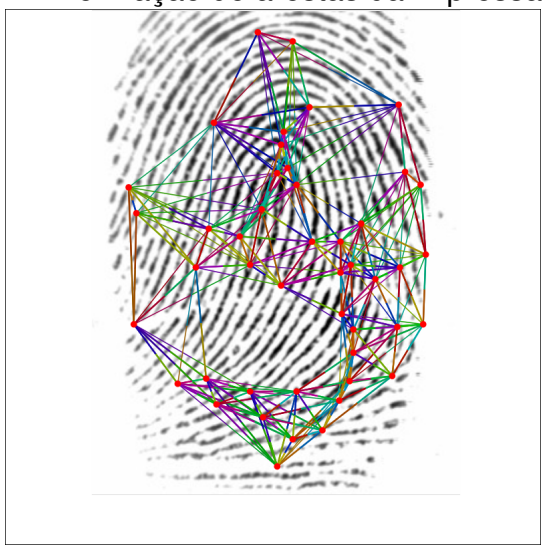
Após a geração das minúcias, há mais um passo de abstração que produz as arestas. Estas são linhas que conectam duas minúcias, e possuem comprimento e dois ângulos que são herdados das minúcias correspondentes. Os dois ângulos são expressos relativos à própria aresta. Estas propriedades não mudam quando a aresta é movida ou rotacionada, e isso é fundamental para a comparação (Važan, 2023).

Figura 10 - Minúcias da impressão digital.

Fonte: Važan (2023).

Após gerar as arestas como demonstrado na Figura 11, o algoritmo então tenta encontrar pelo menos uma aresta compartilhada entre as impressões digitais comparadas. Isso é feito utilizando um algoritmo de vizinho mais próximo com desempenho semelhante à uma tabela *hash*. O resultado disto será o par raíz, sendo este o par inicial de minúcias correspondentes, uma de cada impressão digital. A partir do par raíz, o algoritmo percorre as arestas de dentro para fora e constrói um pareamento composto por vários pares de minúcias e pares de arestas (Važan, 2023).

Figura 11 - Formação de arestas da impressão digital.



Fonte: Važan (2023).

O algoritmo então analisa cuidadosamente o pareamento e decide se tal pareamento significa uma correspondência ou se é apenas uma coincidência. O processo consiste em um sistema de pontuação, a última parte da comparação. Este pressupõe que cada minúcia ou aresta pareada é um evento improvável de acontecer aleatoriamente. Quanto mais destes eventos existirem, menos provável é que o pareamento seja apenas uma coincidência. Assim, o algoritmo essencialmente conta várias características correspondentes e também as pontua com base em quanto elas correspondem. A soma final das pontuações parciais é ajustada para crescer de forma coerente com a similaridade e é retornada pelo algoritmo (Važan, 2023).

A pontuação final é um número aproximado e necessariamente probabilístico. O número não reside em uma escala absoluta de 0 a 100, ele cresce cada vez mais de acordo com a quantidade de minúcias disponíveis e qualidade da imagem. Desta forma, não há como possuir certeza absoluta de uma correspondência independentemente do valor. No entanto, é possível definir um limite razoável, além do qual é suficientemente certo de que as duas impressões digitais correspondem. A certeza no reconhecimento de impressões digitais é medida pelo FMR: taxa de falsos positivos *(false match rate)*. O FMR é a frequência com que o sistema reconhece incorretamente impressões digitais diferentes como sendo correspondentes. Um limite de 40 corresponde a um FMR de 0,01%. As aplicações podem aumentar o limite para obter um FMR exponencialmente menor, ao custo de um pequeno aumento no FNMR: taxa de falsos negativos *(false non-match*

rate) (Važan, 2023). Neste trabalho, o limite escolhido foi 40.

2.3 REALIZAÇÃO DE TESTES CONTROLADOS

Para a validação do sistema desenvolvido, realizaram-se testes em um ambiente controlado utilizando contas de usuário fictícios. O objetivo dos testes é validar a precisão do sistema na aceitação de autenticações legítimas e na rejeição de tentativas inválidas no momento do *login* na forma de pontuação de similaridade da biblioteca SourceAFIS, bem como sua eficiência medida em tempo de execução em segundos. O tempo de execução leva em consideração o tempo da abertura do alerta requisitando a autenticação de segundo fator até a sua completude.

O cenário de teste consistiu na criação de uma conta para o usuário fícticio "Usuário A" e o cadastro de uma impressão digital correspondente ao dedo polegar direito para utilização na autenticação do segundo fator. Em seguida, realizou-se com cada dedo das duas mãos, três tentativas de *login* para validar a pontuação de similaridade, considerando que uma similaridade de 40 ou mais significa uma autenticação legítima. Além deste cenário, realizou-se mais trinta tentativas de *login* com o dedo cadastrado, com intuito de capturar e avaliar a precisão do algoritmo no reconhecimento de autenticações legítimas.

3 DISCUSSÃO E RESULTADOS

O sistema de autenticação de dois fatores desenvolvido se mostrou satisfatório perante seus objetivos. Isto é, o armazenamento das informações biométricas de forma criptografada, bem como a não transferência de dados biométricos por meio da internet. A biblioteca SourceAFIS se mostrou fundamental na execução do projeto, tendo em vista que elimina a necessidade da utilização do algoritmo de comparação de impressões digitais do leitor. Isto é, o processo se tornou seguro contra roubo ou clonagem do leitor, pelo mesmo não possuir qualquer informação.

A utilização de *tokens* de acesso se mostrou prática. Com validade de 15 minutos, eles são atualizados com outro *token*, armazenado em um *cookie* HTTP, com validade de oito horas. Ou seja, eles adicionam uma camada de segurança às requisições da aplicação ao mesmo tempo em que facilitam a vida do usuário deixando-o logado durante o dia caso necessário.

Apesar de não existir transferência dos dados da impressão digital nas requisições HTTP, ainda existe a transmissão dos *tokens* que ga-

rantem a autenticação no sistema. A implementação do protocolo HTTPS *(Hypertext Transfer Protocol Secure)* aprimoraria a segurança do processo, tendo em vista que a utilização do mesmo garantiria a transmissão dos dados de forma criptografada. Este não foi implementado no presente projeto por questão econômica, pois representa um custo adicional na forma de aquisição de um certificado digital e por questão contextual, com o escopo principal deste trabalho sendo a demonstração do funcionamento das funcionalidadaes que tangem à impressão digital.

Os dados coletados para o cenário de teste das tentativas de *login* com a impressão digital de todos os dedos são apresentados na Tabela 1. Foi adicionada a média dos valores de similaridade e tempo de execução das tentativas para análise em todas as tabelas. O tamanho de amostra dos testes realizados foram três para cada dedo da mão, totalizando 30 testes. O número se justifica por ser um ambiente controlado sem variação de usuários diferentes, onde um tamanho de amostra muito maior seria necessário para melhor comparação.

Na Tabela 1, a média da pontuação de similaridade do polegar direito, que equivale ao dedo cadastrado, é de 128,65, número 3,2 vezes maior do que o limite estabelecido de 40. Os outros dedos, que não são válidos para a conta do usuário, atingiram média máxima de 5,38 na forma do anelar esquerdo, número muito inferior ao limite de uma autenticação legítima. É evidente que o algoritmo de comparação é extremamente competente. Ademais, deduz-se que a imagem gerada pelo leitor tem qualidade suficientemente boa para gerar *templates* SourceAFIS capazes de serem comparados com facilidade.

A média de tempo geral para a autenticação do segundo fator foi de 10,33 segundos. Este valor se justifica pelo processo de transferência de dados da imagem da impressão digital do *Hardware* ao Servidor, onde maior parte deste tempo é empregada devido a taxa de transferência do ESP32 ser de 57600 *bits* por segundo. A mudança desta taxa para um maior valor acarretou em erros no código o que representou uma limitação no processo.

Tabela 1 - Dados de tentativas de *login* com todos os dedos.

Dedo	Tentativa 1	Tentativa 2	Tentativa 3	Média de Similaridade	Média de tempo (s)
Polegar Direito	133,43	116,95	135,57	128,65	10,35
Indicador Direito	6,68	0,04	0,92	2,55	10,19
Médio Direito	1,95	0,52	0	0,82	9,76
Anelar Direito	0,45	0,72	0,41	0,53	9,74
Mínimo Direito	2,52	2,48	0	1,67	9,91
Polegar Esquerdo	6,26	0	8,04	4,77	10,58
Indicador Esquerdo	0,81	7,13	6,27	4,74	9,83
Médio Esquerdo	4,91	2,01	0	2,31	10,43
Anelar Esquerdo	10,47	5,28	0,38	5,38	11,37
Mínimo Esquerdo	0	0,7	0	0,23	11,13

Fonte: Elaborado pelo autor.

A Tabela 2 demonstra os dados coletados para o cenário de teste de autenticações legítimas. O tamanho de amostra utilizado foi 30, para igualar o número de testes da Tabela 1. A média da pontuação de similaridade foi de 99,01, o que, apesar de abaixo da média do teste anterior, reforça a precisão do algoritmo e qualidade do sensor empregados. A diversidade dos números de similaridade se explica pela variação de posicionamento e pressão empregado no dedo no momento da inserção no leitor, fatores impossíveis de controlar de forma intencional. O tempo de execução segue a mesma tendência da Tabela 1.

Tabela 2 - Dados de tentativas de *login* com impressão digital correta

Tentativa	Similaridade	Tempo (s)	Tentativa	Similaridade	Tempo (s)
1	144,91	10,95	**16**	85,68	11,29
2	98,56	10,19	**17**	103,11	10,46
3	81,46	10,22	**18**	136,34	11,72
4	91,67	10,19	**19**	97,94	11,34
5	77,32	9,72	**20**	103,6	11,34
6	112,55	10,58	**21**	111,39	11,16
7	129,46	10,6	**22**	86,25	11,49
8	76,65	10,46	**23**	90,29	10,75
9	84,56	11,19	**24**	127,5	12,55
10	114,89	10,85	**25**	108,19	10,74
11	104,62	10,57	**26**	91,56	12,24
12	76,58	10,24	**27**	94,78	10,29
13	100,05	10,63	**28**	85,67	10,9
14	75,58	10,23	**29**	95,21	10,77
15	86,05	10,98	**30**	97,87	11,32
Média	-	-	-	**99,01**	**10,87**

Fonte: Elaborado pelo autor.

Os sistemas de autenticação de dois fatores apresentados por Köhler et al. (2021), em sua maioria, confiam o processo de autenticação

do segundo fator ao celular do usuário e, no caso de um programa apresentado, os dados biométricos têm de ser transmitidos por meio da internet. O presente projeto se distancia fundamentalmente dos artefatos apresentados, retirando a necessidade de confiança em um artefato externo e da transferência de dados biométricos pela rede. Isto torna o processo mais seguro contra ataques de interceptação de dados por exemplo, mas também implica em tempo adicionado ao processo de autenticação, visto que o armazenamento da impressão digital no celular faz com que o processo de comparação seja quase instantâneo.

A arquitetura apresentada por Bian et al. (2020) requer a criação de um identificador pelo próprio usuário e uma chave secreta gerada por sua impressão digital inserida em seu celular para a autenticação do segundo fator. Bian et al. (2020) também utiliza chaves PUF *(Physically Unclonable Function)*, geradas por dispositivos a partir da "impressão digital" do *hardware* do mesmo, teoricamente impossíveis de serem clonadas. É um sistema que funciona sem a utilização do modelo tradicional de usuário e senha, divergente do presente artigo que necessita do *token* gerado pelo *login* na conta com a senha correta. O presente projeto não requer outro dispositivo de *hardware* que não o leitor de impressões digitais para a autenticação do segundo fator, apesar de necessitar da instalação do servidor de autenticação local.

Segoro e Putro (2020) apresentam um esquema que combina diversos métodos de autenticação de segundo fator em uma solução, sendo estes *QR Codes*, *OTP* e impressões digitais. O presente projeto emprega apenas uma técnica de autenticação para o segundo fator, o que pode ser considerado mais amigável e prático para o usuário. Entretanto, a aplicação do artigo de Segoro e Putro (2020) é divergente deste, o que pode prejudicar a comparação de usabilidade.

O sistema de autenticação de segundo fator apresentado por Shah e Kanhere (2019) funciona por meio da captura da perturbação causada no sinal Wi-Fi pela atividade cardiopulmonar dos indivíduos. A natureza experimental do artigo o diferencia do presente projeto ao sugerir um método de autenticação completamente novo, com precisão máxima de 65% em grupos de cinco pessoas. Apesar disso, o mesmo conta com testes de voluntários, ausentes neste artigo.

Kamiński et al. (2023) apresenta um sistema de autenticação de dois fatores utilizando a voz dos indivíduos. O sistema possui o processo tradicional de usuário e senha e, para autenticar o segundo fator requer o

cadastro de um clipe de 25 segundos da voz do usuário. No momento de acesso ao sistema, o indivíduo necessita submeter um clipe de 5 segundos para comparação. O tempo de execução e a atipicidade do processo pode causar problemas de usabilidade para os usuários. A ação de inserir a impressão digital em um leitor apresentada neste artigo se demonstra mais simples e rápida. Entretanto, o trabalho de Kamiński et al. (2023) possui um extenso teste com voluntários, somando 55 pessoas e atingindo 91,82% de precisão.

4 CONCLUSÃO

O presente trabalho teve como objetivo a construção de um sistema de autenticação de dois fatores utilizando biometria. O mesmo foi atingido com o desenvolvimento de artefatos de *software* e *hardware*. O sistema demonstrou proporcionar segurança aprimorada contra tipos de vulnerabilidades comuns, como ataques de força bruta e violações de dados.

A principal limitação do sistema desenvolvido foi o tempo demandado para a autenticação da impressão digital em função do processo de transferência dos dados do leitor ao servidor de autenticação. O tempo elevado se justifica pela não utilização do processo de autenticação interno do leitor, medida que trocou maior segurança por menos eficiência neste processo. No entanto, é possível estudar e implementar maneiras para diminuição deste tempo.

Os resultados dos testes controlados demonstraram a precisão elevada do algoritmo utilizado para a comparação de impressões digitais no momento de acesso ao sistema. Entretanto, é essencial a realização de testes com grupos de voluntários para validar de forma mais eficaz a usabilidade e precisão do sistema em condições reais de uso.

Adicionalmente, pode-se estudar a implementação do sistema em um ambiente nuvem *(cloud)* utilizando tecnologias como o MQTT *(Message Queuing Telemetry Transport)*, que podem auxiliar a manter ainda a propriedade de não transferência dos bytes biométricos pela rede, dado que uma estrutura adequada seja montada.

Como trabalhos futuros sugere-se: estudar e implementar maneiras de diminuir o tempo da autenticação; submeter o sistema a testes com grupos de voluntários; submeter o sistema a testes de outros tipos de ataques cibernéticos; estudar maneiras de implementar o sistema em ambiente nuvem.

REFERÊNCIAS

BCRYPT. **bcrypt - npm**. 2023. Acesso em: 21 abr. 2025. Disponível em: <https://www.npmjs.com/package/bcrypt>.

BERRIOS, J. et al. **Factorizing 2FA: Forensic analysis of two-factor authentication applications**. [s.n.], 2023. v. 45. 301569 p. ISSN 2666-2817. Disponível em: <https://www.sciencedirect.com/science/book/pii/S2666281723000781>.

BIAN, W. et al. **Bio-AKA: An efficient fingerprint based two factor user authentication and key agreement scheme**. [s.n.], 2020. v. 109. 45-55 p. ISSN 0167-739X. Disponível em: <https://www.sciencedirect.com/science/book/pii/S0167739X19332467>.

DAS, S.; WANG, B.; CAMP, L. J. **Mfa is a waste of time! understanding negative connotation towards mfa applications via user generated content**. [S.l.: s.n.], 2019.

DUTSON, J. et al. **Don't Punish all of us: Measuring User Attitudes about Two-Factor Authentication**. [S.l.: s.n.], 2019. 119-128 p.

FARKE, F. M. et al. **"You still use the password after all" – Exploring FIDO2 Security Keys in a Small Company**. USENIX Association, 2020. 19–35 p. ISBN 978-1-939133-16-8. Disponível em: <https://www.usenix.org/conference/soups2020/presentation/farke>.

GILSENAN, C. et al. **Security and Privacy Failures in Popular** {**2FA**} **Apps**. [S.l.: s.n.], 2023. 2079–2096 p.

GOLLA, M. et al. **Driving 2FA Adoption at Scale: Optimizing Two-Factor Authentication Notification Design Patterns**. USENIX Association, 2021. 109–126 p. ISBN 978-1-939133-24-3. Disponível em: <https://www.usenix.org/conference/usenixsecurity21/presentation/golla>.

GUPTA, A. et al. **Two-Factor Authentication Using QR Code and OTP**. [S.l.], 2023. 105–114 p.

KAMIńSKI, K. A. et al. **Enhancing Web Application Security: Advanced Biometric Voice Verification for Two-Factor Authentication**. [s.n.], 2023. v. 12. ISSN 2079-9292. Disponível em: <https://www.mdpi.com/2079-9292/12/18/3791>.

KOISTINEN, M. et al. **Tolerance for typographical errors on password authentication securely via keystroke dynamics**. Dissertação (Mestrado), 2021.

KöHLER, D. et al. **Assessment of Remote Biometric Authentication Systems: Another Take on the Quest to Replace Passwords**. [S.l.: s.n.], 2021. 22-31 p.

MANSOUR, A. et al. **A Lightweight Seamless Unimodal Biometric Authentication System**. [s.n.], 2024. v. 231. 190-197 p. 14th International Conference on Emerging Ubiquitous Systems and Pervasive Networks / 13th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (EUSPN/ICTH 2023). ISSN 1877-0509. Disponível em: <https://www.sciencedirect.com/science/book/pii/S1877050923022044>.

OESCH, S.; RUOTI, S. **That Was Then, This Is Now: A Security Evaluation of Password Generation, Storage, and Autofill in Browser-Based Password Managers**. USENIX Association, 2020. 2165–2182 p. ISBN 978-1-939133-17-5. Disponível em: <https://www.usenix.org/conference/usenixsecurity20/presentation/oesch>.

PATHAN, A. et al. **Fingerprint authentication security: An improved 2-step authentication method with flexibility**. [S.l.: s.n.], 2019. v. 10.

PRATAMA, A. R.; FIRMANSYAH, F. M. **Until you have something to lose! Loss aversion and two-factor authentication adoption**. Emerald Publishing Limited, 2021. ahead-of-print. Disponível em: <https://doi.org/10.1108/ACI-12-2020-0156>.

REESE, K. et al. **A Usability Study of Five Two-Factor Authentication Methods**. Santa Clara, CA: USENIX Association, 2019. 357–370 p. ISBN 978-1-939133-05-2. Disponível em: <https://www.usenix.org/conference/soups2019/presentation/reese>.

SEGORO, M. B.; PUTRO, P. A. W. **Implementation of Two Factor Authentication (2FA) and Hybrid Encryption to Reduce the Impact of Account Theft on Android-Based Instant Messaging (IM) Applications**. [S.l.: s.n.], 2020. 115-120 p.

SHAH, S. W.; KANHERE, S. S. **Smart user identification using cardiopulmonary activity**. [s.n.], 2019. v. 58. 101024 p. ISSN 1574-1192. Disponível em: <https://www.sciencedirect.com/science/book/pii/S1574119218305145>.

VAžAN, R. **How SourceAFIS algorithm works**. 2023. Acesso em: 26 abr. 2025. Disponível em: <https://sourceafis.machinezoo.com/algorithm>.