Sistema Biométrico baseado em Eletrocardiografia e Impressão Digital

JOAO RICARDO VIDEIRA CUSTODIO

(Licenciado em Engenharia Eletrotécnica)

Dissertação para obtenção de grau de Mestre em Engenharia Biomédica

Orientadores:
Doutor André Ribeiro Lourenço
Doutor Hugo Plácido da Silva

Júri:
Presidente: Doutora Cecília Ribeiro da Cruz Calado
Vogais:
Doutor Artur Jorge Ferreira
Doutor André Ribeiro Lourenço

Setembro 2025

Sistema Biométrico baseado em Eletrocardiografia e Impressão Digital

JOAO RICARDO VIDEIRA CUSTODIO

(Licenciado em Engenharia Eletrotécnica)

Dissertação/Trabalho de Projeto/Relatório de Estágio para obtenção de grau de Mestre em Engenharia Biomédica

Orientadores:
Doutor André Ribeiro Lourenço, ISEL
Doutor Hugo Plácido da Silva, IT

Júri:
Presidente: Doutora Cecília Ribeiro da Cruz Calado
Vogais:
Doutor Artur Jorge Ferreira
Doutor André Ribeiro Lourenço

Setembro 2025

Agradecimentos

Quero agradecer aos Professores André Lourenço e Hugo Plácido da Silva pela orientação, disponibilidade e apoio ao longo de todo este percurso. O vosso acompanhamento foi decisivo para que esta dissertação se tornasse realidade. Agradeço também ao Instituto de Telecomunicações e à CardioID pelo apoio prestado ao desenvolvimento deste trabalho. Expresso igualmente o meu agradecimento ao projeto IPL/IDI&CA2024/M-IA-RCH_ISEL, cujo apoio foi fundamental para a concretização desta dissertação.

Aos meus amigos, Patrícia, Ricardo, Diogo e André agradeço pelos momentos de convívio e descontração, que, mais do que simples pausas, foram essenciais para renovar energias e manter a motivação ao longo deste percurso. Quero ainda deixar uma palavra de agradecimento à Filipa, ao Pedro e ao Rúben, pelos bons momentos partilhados, em especial nas longas sessões de jogos, que ajudaram a tornar esta etapa mais leve.

Por fim, deixo um agradecimento muito especial a toda a minha família, em particular à minha mãe, ao meu irmão Miguel e aos meus avós, Gilda e João pelo apoio incondicional, pela compreensão e pelo incentivo constantes. Sem o vosso suporte, esta jornada teria sido incomparavelmente mais difícil.

Declaração de integridade

Declaro que esta dissertação é o resultado da minha investigação pessoal e independente. O seu conteúdo é original e todas as fontes listadas nas referências bibliográficas foram consultadas e estão devidamente mencionadas no texto. Mais declaro que todas as referências científicas e técnicas relevantes para o desenvolvimento do trabalho estão devidamente citadas e constam das referências bibliográficas.

Adicionalmente, declaro que recorri a ferramentas de apoio baseadas em Modelos de Linguagem de Grande Escala (LLMs), nomeadamente o ChatGPT (OpenAI), para efeitos de apoio na redação, revisão de texto e desenvolvimento de código, mantendo sempre a responsabilidade integral pelo conteúdo científico, técnico e pelas conclusões apresentadas.

João Custódio

Lisboa, 28 de setembro de 2025

Resumo

O reconhecimento biométrico estabeleceu-se como uma alternativa mais segura em relação a métodos tradicionais como palavras-passe, cartões ou *tokens* de utilização pessoal. Contudo os sistemas unimodais apresentam algumas falhas significativas no que diz respeito à taxa de falsas aceitações, presença de ruído e suscetibilidade a tentativas de falsificação. Embora os sistemas biométricos multimodais ofereçam diversas vantagens relativamente a estes anteriores, a sua adoção ainda é limitada uma vez que se estima que apenas cerca de 30% dos sistemas de autenticação utilizem duas ou mais modalidades biométricas. Esta dissertação propõe a utilização de uma abordagem bimodal, combinando dois sistemas biométricos distintos para melhorar a precisão e segurança do utilizador.

O primeiro sistema baseia-se na impressão digital, uma das formas mais comuns e amplamente utilizadas de identificação biométrica. O segundo sistema consiste numa modalidade de biometria que tem o potencial de complementar as abordagens existentes devido à sua natureza intrínseca, a utilização do eletrocardiograma (ECG) como fonte biométrica. O sistema desenvolvido inclui a aquisição automática dos sinais de ECG, extração de minúcias a partir das imagens de impressões digitais, e um módulo de fusão a nível de score que pondera os contributos de ambas as modalidades. A implementação foi realizada em Python num Raspberry Pi 5, com suporte para registo, autenticação e identificação de utilizadores. Foram realizados testes experimentais com um conjunto de 12 participantes, avaliando métricas como a Taxa de Erro Igual (EER), a Taxa de Aceitação Indevida (FAR) e a Taxa de Rejeição indevida (FRR).

Os resultados demonstraram que essa abordagem atingiu um melhor desempenho quando comparada com as metodologias de forma isolada, reduzindo significativamente as taxas de erro e aumentando a robustez perante potenciais ataques de *spoofing*. A integração deste sistema mostrou-se uma solução viável e promissora para sistemas de autenticação biométrica de baixo custo.

Palavras-chave: impressão digital, eletrocardiograma, biometria, multimodal.

Abstract

Biometric recognition is currently established as a more secure alternative to traditional methods such as passwords, cards or personal tokens. However, unimodal systems have some significant flaws in terms of false acceptance rates, noise presence and susceptibility to forgery attempts. Although multimodal biometric systems offer several advantages over the former, their adoption is still limited, as it is estimated that only about 30% of authentication systems use two or more biometric modalities. This dissertation proposes the use of a bimodal approach, combining two distinct biometric systems to improve accuracy and user security.

The first system is based on fingerprinting, one of the most common and widely used forms of biometric identification. The second system consists of a biometric modality that has the potential to complement existing approaches due to its intrinsic nature, the use of the electrocardiogram (ECG) as a biometric source. The developed system includes automatic acquisition of ECG signals, extraction of minutiae from fingerprint images, and a score-level fusion module that weighs the contributions of both modalities. The implementation was carried out in Python on a Raspberry Pi 5, with support for user registration, authentication, and identification. Experimental tests were carried out with a group of 12 participants, evaluating metrics such as Equal Error Rate (EER), False Acceptance Rate (FAR) and False Rejection Rate (FRR).

The results showed that this approach achieved better performance when compared to the methodologies in isolation, significantly reducing error rates and increasing robustness against potential spoofing attacks. The integration of this system proved to be a viable and promising solution for low-cost biometric authentication systems.

Keywords: fingerprint, electrocardiogram, biometrics, multimodal.

Lista de Símbolos e de Siglas

SSG *Soil Stiffness Gauge*

AFIS *Automated Fingerprint Identification System*

AUC *Area Under Curve*

CCD *Charge-Coupled Device*

CMOS *Complementary Metal–Oxide–Semiconductor*

CNN *Convolutional Neural Network*

CSV *Comma-Separated Values*

ECG *Eletrocardiograma*

EER *Equal Error Rate*

FAR *False Acceptance Rate*

FMR *False Match Rate*

FNMR *False Non-Match Rate*

FTA *Failure To Acquire*

FTE *Failure To Enroll*

FRR *False Rejection Rate*

FTDI *Future Technology Devices International Limited*

FTIR *Failure To Identify Rate*

IEC *International Electrotechnical Commission*

ISO *International Organization for Standardization*

K-NN *K-Nearest Neighbors*

NIST *National Institute of Standards and Technology*

LLM *Large Language Models*

PLI *Powerline Interference*

RBF *Radial Basis Function*

ROC *Receiver Operating Characteristic*

SIFT *Scale-Invariant Feature Transform*

SVM *Support Vector Machine*

TTL *Transistor-Transistor Logic*

USB *Universal Serial Bus*

WDIST *Wavelet Distance*

Índice

Índice de Figuras

Índice de Tabelas

´

1. Introdução

1.1 Enquadramento

A biometria é a área que estuda o reconhecimento de um indivíduo com base em atributos físicos ou comportamentais, como impressões digitais, íris, voz, ou assinatura. Esta é fundamental para a segurança e acesso dos indivíduos a sistemas em substituição a métodos mais tradicionais como a utilização de senhas de acesso ou *tokens* de utilização pessoal [1]. Através de características biológicas é possível mitigar os riscos associados a fraude ou até mesmo furto de informação. Dentro da biometria existem os dados biométricos fisiológicos e também os comportamentais, sendo que atualmente o dado biométrico mais utilizado em instituições públicas [2] é a impressão digital, que se insere na categoria dos fisiológicos [3]. Isto prende-se no facto de ser de fácil identificação, intransmissível, e dificilmente falsificado.

Apesar da biometria ser geralmente segura, dada a sua natureza intrínseca e o armazenamento dos dados de forma encriptada [2], esse sistema pode falhar em determinadas situações. Podem ocorrer *Presentation Attacks,* nos quais um atacante utiliza características físicas ou dados biométricos de outra pessoa para aceder a um determinado sistema ou plataforma [3], [4]. Estes ataques afetam todas as modalidades biométricas e têm sido objeto de atenção significativa pela comunidade científica e técnica, resultando na criação de normas como uma solução mais robusta, que torna o processo de identificação e autenticação mais seguro e difícil de contornar, como os sistemas de autenticação de dois fatores nos quais o telemóvel tem sido incrementalmente utilizado como modalidade adicional de segurança. Ao combinar diferentes características biométricas, o sistema aumenta a fiabilidade e reduz a probabilidade de falhas ou tentativas de falsificação, proporcionando assim uma proteção mais eficaz contra acessos não autorizados [5].

Como resultado, alcança-se uma Taxa de Aceitação Indevida (FAR) e de Rejeição Indevida (FRR) substancialmente mais baixa em comparação com sistemas que utilizam apenas uma modalidade biométrica de forma isolada, como é possível observar na Tabela 1 [2].

Tabela 1 – Taxas de Rejeição Indevida (FRR) e Aceitação Indevida (FAR) no sistema de reconhecimento por ECG e impressão digital (adaptado de [2]).

Biometrias	*FAR (%)*	*FRR (%)*
Impressão Digital	7.77	5.55
Eletrocardiograma	2.38	9.52
Fusão	2.5	0

No âmbito desta dissertação, merece destaque a empresa portuguesa CardioID Technologies, que tem desenvolvido soluções biométricas assentes em ECG, entre as quais se encontra o dispositivo CardioWheel [6], utilizado também neste trabalho.[1]

1.2 Definição do Problema

Apesar dos avanços na área, os sistemas unimodais continuam a apresentar limitações críticas em ambientes de elevada segurança, nomeadamente no que diz respeito à precisão, praticabilidade, custo e também à suscetibilidade a *Presentation Attacks* e tentativas de falsificação. Além disso, a dependência exclusiva de uma única característica biométrica compromete a fiabilidade em casos de lesão, desgaste ou variações fisiológicas. Surge assim a necessidade de desenvolver sistemas biométricos mais seguros e adaptáveis, que combinem mais do que uma fonte de informação para garantir uma autenticação mais robusta [7], [8].

1.3 Objetivos

Este trabalho propõe um sistema biométrico bimodal que combina sinais de ECG (eletrocardiograma) com impressões digitais, explorando a complementaridade entre

[1] Mais informações disponíveis em https://www.cardio-id.com

uma modalidade fisiológica interna (ECG) e uma externa (impressão digital). As principais contribuições desta tese incluem:

- A revisão crítica do estado da arte em autenticação biométrica unimodal;
- O desenvolvimento de uma sequência de etapas para aquisição e processamento de dados biométricos de ECG e impressões digitais;
- A análise da complementaridade entre ambas as modalidades para fins de autenticação e identificação;
- A avaliação experimental do desempenho do sistema proposto, com métricas como FAR, FRR e EER.

1.4 Estrutura da Tese

Esta dissertação encontra-se organizada da seguinte forma:

- **Capítulo 1 – Introdução**: Apresenta o contexto, o problema, os objetivos e a estrutura da tese;
- **Capítulo 2 – Fundamentos Teóricos**: Descreve os conceitos e fundamentos teóricos necessários à compreensão do trabalho.
- **Capítulo 3 – Estado de Arte**: Analisa estudos realizados sobre o tema e soluções já existentes identificando lacunas que justifiquem o desenvolvimento do presente trabalho.
- **Capítulo 4 – Materiais e Métodos**: Detalha toda a metodologia utilizada no desenvolvimento do presente trabalho.
- **Capítulo 5 – Resultados**: Apresenta os resultados obtidos, tanto ao nível da implementação como da validação experimental, acompanhados de uma análise crítica do desempenho do sistema e comparação com a literatura existente.
- **Capítulo 6 – Conclusão e Trabalho Futuro**: Resume os principais contributos da dissertação, reflete sobre os objetivos alcançados e propõe possíveis direções para trabalhos futuros com base nas limitações e oportunidades identificadas.

2. Fundamentos Teóricos

Esta secção apresenta os fundamentos teóricos necessários para a compreensão do trabalho desenvolvido. Inicia-se com uma descrição geral da biometria, destacando os princípios e etapas fundamentais dos sistemas biométricos. De seguida, é abordada a modalidade da impressão digital, com enfoque nos sensores utilizados e nos principais métodos de reconhecimento. Posteriormente, analisa-se o eletrocardiograma (ECG) como sinal biométrico, explorando as suas características fisiológicas e os métodos de reconhecimento baseados neste tipo de informação. O objetivo é fornecer uma visão estruturada dos conceitos essenciais que suportam a implementação e validação do sistema biométrico multimodal desenvolvido nesta dissertação.

2.1 Biometria

Os sistemas biométricos atuais são constituídos por duas fases importantes que são o registo em que o utilizador insere os seus dados numa interface e a autenticação ou identificação que é quando esses dados são comparados com um ou mais utilizadores.

Na fase de registo, como exemplificado na Figura 1, o utilizador fornece uma amostra biométrica, como por exemplo uma impressão digital, um sinal de ECG ou outra característica física ou comportamental. Essa amostra é captada por um sensor, processada e convertida num modelo digital, designado por *template*, que é posteriormente armazenado numa base de dados segura. Mais tarde, durante a fase de autenticação (verificação) ou identificação, o sistema volta a captar uma nova amostra biométrica e compara-a com o *template* registado.

No caso da autenticação (Figura 2) o objetivo é verificar se a amostra corresponde à identidade que o utilizador afirma ter, realizando uma comparação um-para-um (1:1). Já na identificação (Figura 3) o sistema procura determinar quem é o utilizador comparando a amostra com todos os *templates* existentes na base de dados, o que corresponde a uma comparação um-para-muitos (1:N).

O sucesso e a fiabilidade destas operações dependem da qualidade da aquisição, da robustez dos algoritmos de extração e correspondência, e da segurança com que os dados biométricos são tratados ao longo do processo [9], [10].

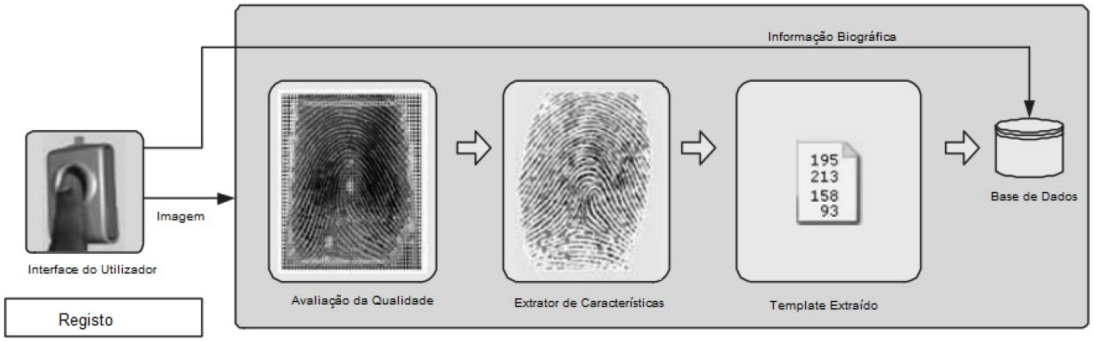

Figura 1 – Processo de registo de impressão digital (adaptado de [10]).

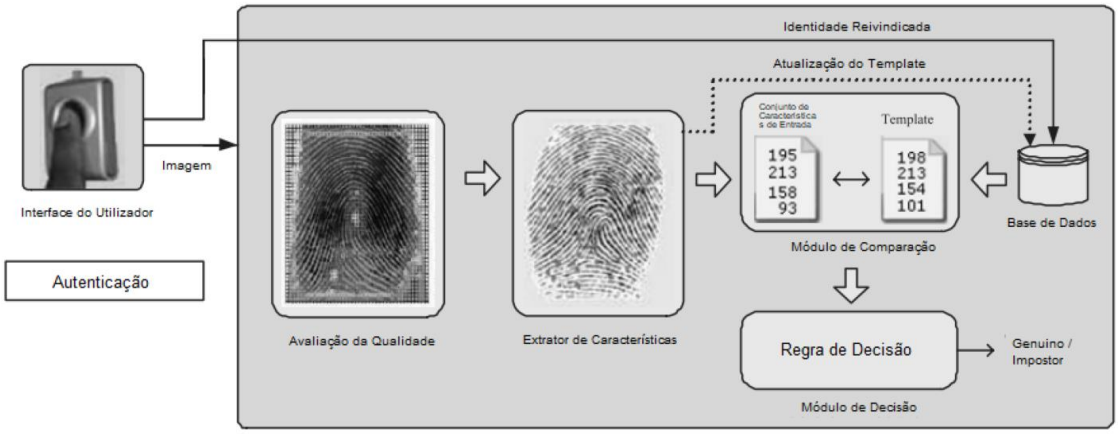

Figura 2 – Processo de autenticação de impressão digital (adaptado de [10]).

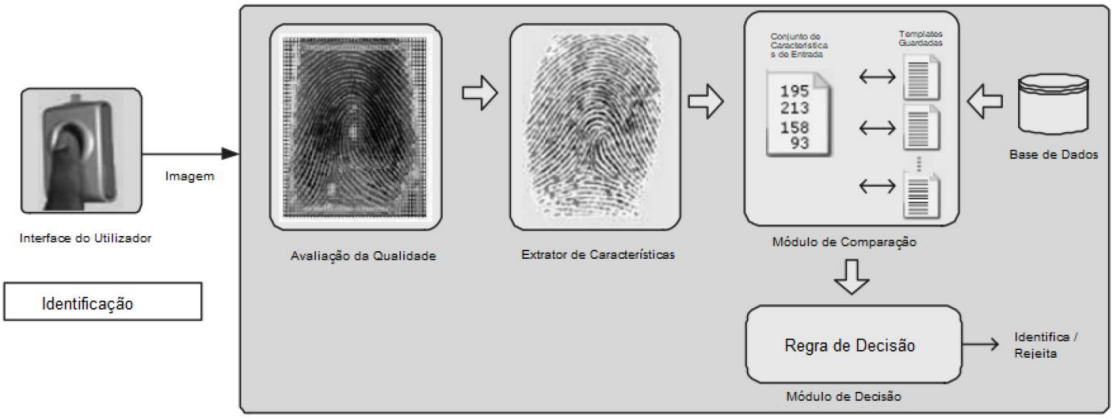

Figura 3 – Processo de identificação de impressão digital (adaptado de [10]).

Existem diversas métricas utilizadas para avaliar o desempenho de sistemas biométricos, sendo as mais comuns a Taxa de Correspondência Falsa (FMR) e a Taxa de Não-Correspondência Falsa (FNMR). A FMR, também designada por Taxa de Aceitação Indevida (FAR), corresponde à probabilidade de o sistema aceitar incorretamente um impostor como se fosse um utilizador legítimo. Por sua vez, a FNMR, igualmente conhecida por Taxa de Rejeição Indevida (FRR), representa a probabilidade de rejeitar indevidamente um utilizador genuíno. A combinação destas duas métricas origina a Taxa de Erro Igual (EER), um indicador amplamente utilizado para comparar algoritmos e sistemas de autenticação biométrica. Quanto menor for o valor da EER, melhor será o desempenho global do sistema (ver Figura 4). A EER corresponde ao ponto da Curva Característica de Operação do Recetor (ROC) em que a Taxa de Aceitação Indevida e a Taxa de Rejeição Indevida são iguais (ver Figura 5) [11].

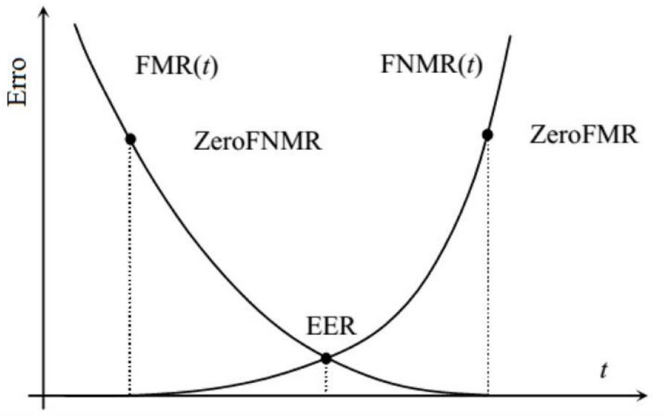

Figura 4 – Curva do ERR (extraído de [11]).

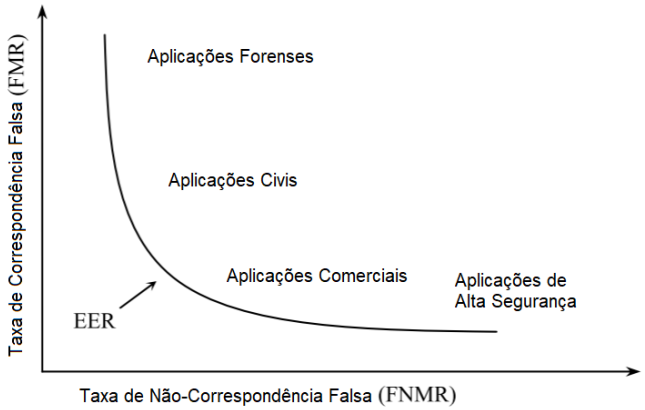

Figura 5 – Curva do ROC (extraído de[11]).

Outros indicadores incluem a Taxa de Falha na Aquisição (FTA) e a Taxa de Falha no Registo (FTE), que refletem problemas na recolha ou registo do sinal biométrico devido a baixa qualidade ou dificuldades de interação com o sensor.

Estas métricas são essenciais não apenas para comparar algoritmos em ambiente controlado, mas também para avaliar a viabilidade de sistemas biométricos em cenários reais, onde a variabilidade do utilizador e dos sensores introduz desafios significativos à robustez e escalabilidade. Estes desafios tornam-se ainda mais evidentes quando se consideram as diferentes modalidades biométricas, ilustradas na Figura 6, que incluem impressão digital, rosto, íris, geometria da mão, padrões da palma, forma da orelha, retina e assinatura.

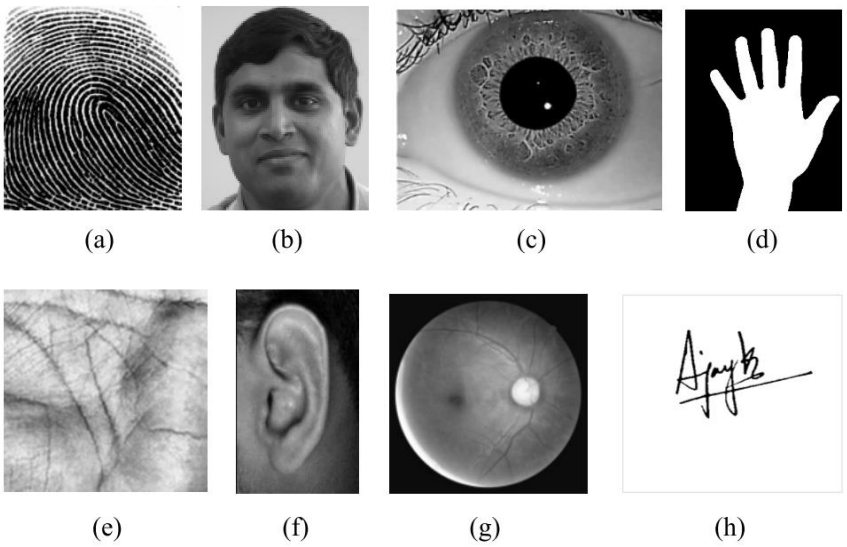

Figura 6 – Traços biométricos normalmente utilizados: (a) impressão digital; (b) rosto; (c) íris; (d) geometria da mão; (e) padrão das veias; (f) orelha; (g) retina; (h) assinatura (extraído de [9]).

2.2 Impressão Digital

As impressões digitais são padrões únicos presentes nas falanges distais dos dedos, caracterizados por uma rede complexa de cristas e vales, representado na Figura 7. Sendo que estes padrões se começam a formar antes do nascimento e permanecem inalterados durante a vida adulta [11] [12].

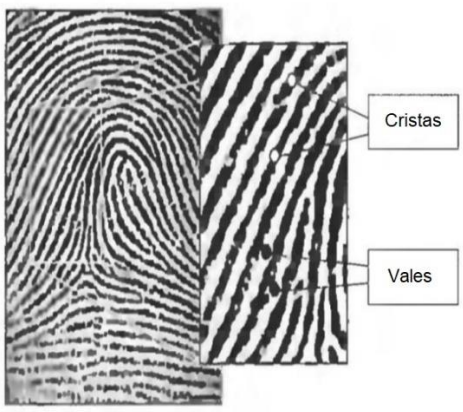

Figura 7 – Constituição de uma impressão digital (adaptado de [13]).

Atualmente existem muitas aplicações para sistemas de reconhecimento de impressões digitais como acesso a dispositivos móveis, monitorização da presença de

funcionários, investigações forenses e criminais entre outras [14], [15]. A popularidade desta tecnologia continua em crescimento devido à sua elevada precisão, praticidade e segurança. Além disso, os avanços tecnológicos têm impulsionado a sua evolução [16].

De acordo com um estudo realizado no *Center for Identity* da Universidade do Texas (Figura 8), diferentes tipos de biometria são amplamente utilizados em instituições públicas e privadas, sendo a impressão digital a tecnologia mais adotada, aplicada em contextos bastante variados [17].

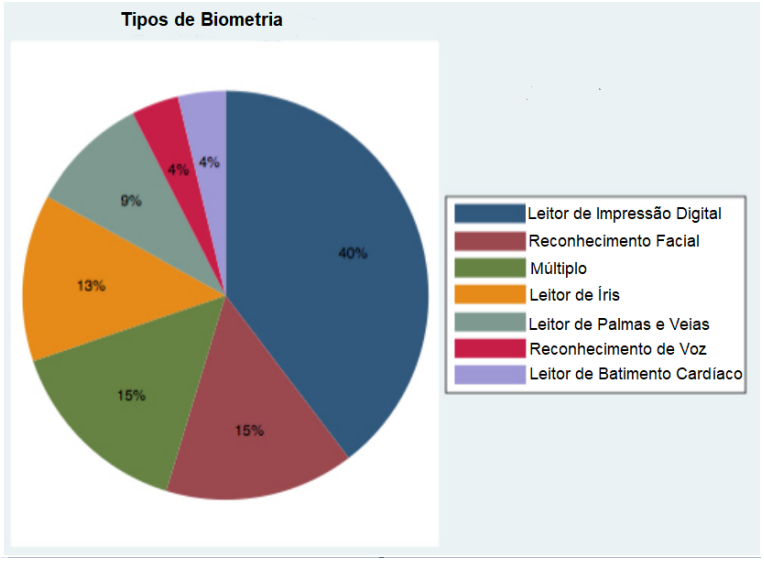

Figura 8 – *Tipos de biometria mais utilizados (adaptado de [17]).*

2.2.1 Sensores de Impressão Digital

Os sensores de impressão digital são dispositivos essenciais em sistemas de reconhecimento biométrico, que utilizam as características únicas da pele e do dedo para identificar e autenticar indivíduos. Atualmente, os três tipos de sensores mais utilizados são os óticos, os capacitivos e os ultrassónicos, cada um com princípios de funcionamento distintos. A escolha do sensor mais adequado depende da aplicação específica e dos requisitos do sistema [14], [16].

Sensores Óticos

Entre os mais conhecidos estão os sensores óticos, que operam com base nos princípios de refração e reflexão da luz para gerar imagens. O processo começa quando o dedo entra em contacto com um prisma de vidro, onde as cristas da impressão digital absorvem a luz e os vales a refletem, criando o padrão único da impressão com precisão. A luz refletida é direcionada para um sensor do tipo Dispositivo de Carga Acoplada (CCD) ou Semicondutor Complementar de Óxido de Metal (CMOS), responsável por converter os sinais luminosos em sinais elétricos e, assim, gerar uma imagem digital detalhada da impressão digital, como está representado na Figura 9 [14], [16], [18].

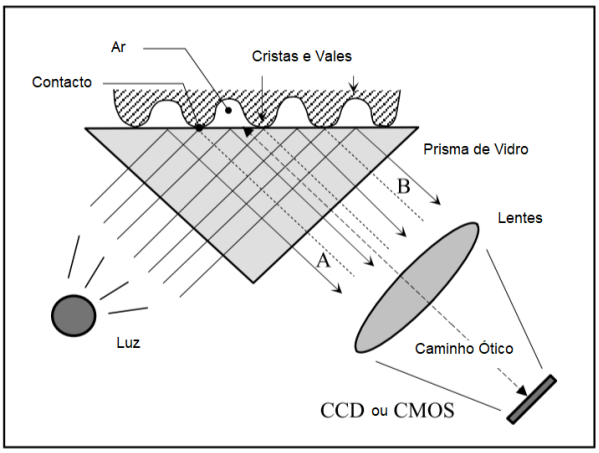

Figura 9 – Método de funcionamento de um sensor ótico FTIR (adaptado de [11]).

Os sensores óticos são conhecidos por serem de baixo custo, mas apresentam algumas limitações. Entre elas, o Processo de Reflexão Total Interna (FTIR), amplamente utilizado nesses sensores, torna difícil miniaturizar o dispositivo, o que limita a sua portabilidade. Além disso, são sensíveis à contaminação na superfície dos dedos ou do sensor, o que pode impactar a qualidade da imagem em dedos secos ou molhados e requerem um grande consumo de energia [16], [19].

Já existem novas tecnologias para melhorar o desempenho destes sensores e mitigar os aspetos negativos, mas atualmente o custo destas melhorias é demasiado elevado comparado ao custo de sensores capacitivos [19].

Sensores Capacitivos

Este tipo de sensores tem vindo a substituir os sensores óticos uma vez que conseguem ter um tamanho mais reduzido e uma menor dependência energética o que os torna mais comuns a utilizar em dispositivos portáteis como telemóveis e computadores.

O seu método de funcionamento consiste na existência de uma matriz de pequenos condensadores ou elétrodos condutores, que em contacto com a pele, medem a capacitância de acordo com os vales e cristas detetados, como representado na Figura 10. Estas diferenças de capacitância são depois convertidas em sinais elétricos. Quanto maior for o contraste da impressão digital mais perto o dedo se encontra da superfície do sensor [14], [16].

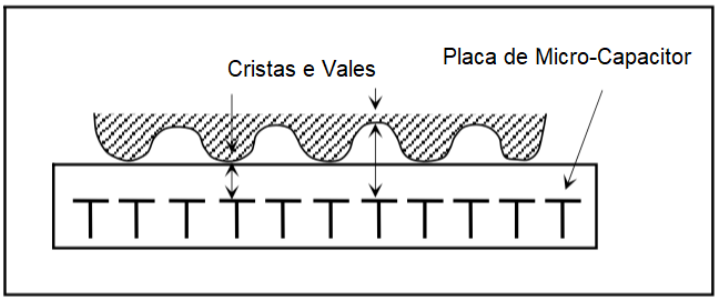

Figura 10 – Método de funcionamento de um sensor capacitivo (adaptado de [11]).

Embora ofereçam várias vantagens, esses sensores apresentam limitações quando usados com dedos excessivamente secos, o que pode comprometer seu desempenho. Por isso, continua-se a explorar alternativas mais avançadas, como os sensores ultrassónicos.

2.2.2 Métodos de Reconhecimento de Impressões Digitais

Atualmente existem diversas metodologias de reconhecimento de impressões digitais, tradicionalmente classificadas em três categorias principais: *correlation-based*, *minutiae-based*, e *ridge feature-based* [20]. Mais recentemente, também têm surgido abordagens baseadas em *deep learning*, que exploram redes neuronais para extrair e comparar características de forma automática.

Os métodos de *correlation-based* identificam e selecionam pequenas regiões na impressão digital primária, sendo que estas regiões ou *templates* contêm informações como cristas, vales e texturas únicas. De seguida os *templates* são comparados com uma segunda imagem calculando o seu nível de semelhança com a imagem de teste através de correlação de níveis de cinzento. Por último são comparadas as posições das *templates* de cada imagem para ver se coincidem de uma forma consistente para a posterior tomada de decisão. Estes métodos apresentam uma solução relativamente simples uma vez que não é necessário a realização de um grande pré-processamento das imagens como nos métodos de *minutiea-based*, diminuindo assim o erro durante essas etapas [21].

Já os métodos de *minutiae-based* são os mais amplamente utilizados, uma vez que são extraídos pontos de minúcia como terminações e bifurcações de duas impressões digitais e são posteriormente armazenadas em vetores tridimensionais (contendo as coordenadas de localização e a orientação dos ângulos) [22]. É depois encontrado um alinhamento entre esses pontos de uma impressão para a outra para ser possível a respetiva comparação [11]. Este é o método mais amplamente utilizado devido à sua elevada precisão e à baixa exigência de recursos computacionais. No entanto, embora seja confiável, a extração de minúcias pode tornar-se imprecisa em imagens de baixa qualidade, o que motivou a investigação de abordagens alternativas [20].

Por sua vez, os métodos *ridge-based* são uma abordagem que utiliza características das cristas das impressões digitais como direção, orientação, espessura, frequência e padrões para análise e sucessiva comparação. Essa abordagem pode complementar ou substituir os métodos baseados em minúcias, sendo particularmente eficaz para imagens de baixa qualidade ou para sensores compactos que capturam apenas uma parte limitada da impressão digital [11], [23].

Os métodos de *deep learning* utilizam Redes Neuronais Convolucionais (CNNs) para extrair automaticamente características relevantes das imagens, como padrões de cristas e bifurcações. O processo começa com a aquisição e pré-processamento das imagens, seguido pela sua introdução num modelo treinado para classificar ou comparar impressões. Para tarefas de verificação, utilizam-se redes siamesas, capazes de comparar pares de impressões digitais e medir a sua similaridade. Na extração de minúcias, recorrem-se a redes convolucionais especializadas que aprendem automaticamente a localizar e orientar os pontos característicos da

impressão. Estes métodos apresentam elevada precisão (acima de 99%) e maior robustez face a variações de qualidade, iluminação e rotação [24], [25].

2.3 Eletrocardiograma

O eletrocardiograma (ECG) constitui uma das ferramentas mais consolidadas na prática médica para a monitorização da atividade elétrica cardíaca. Trata-se de um sinal bioelétrico gerado pela despolarização e repolarização sequencial das fibras musculares cardíacas, refletindo os potenciais de ação que se propagam a partir do sistema de condução elétrica do coração [26], [27].

A morfologia típica do ECG é composta por cinco deflexões principais: as ondas P, Q, R, S e T. A onda P reflete a despolarização atrial, o complexo QRS corresponde à rápida despolarização ventricular, e a onda T está relacionada com a repolarização dos ventrículos, como ilustrado na Figura 11. Em alguns casos, observa-se também uma onda U, cuja origem ainda é debatida, mas pode estar associada à repolarização do sistema de Purkinje [27], [28], [29].

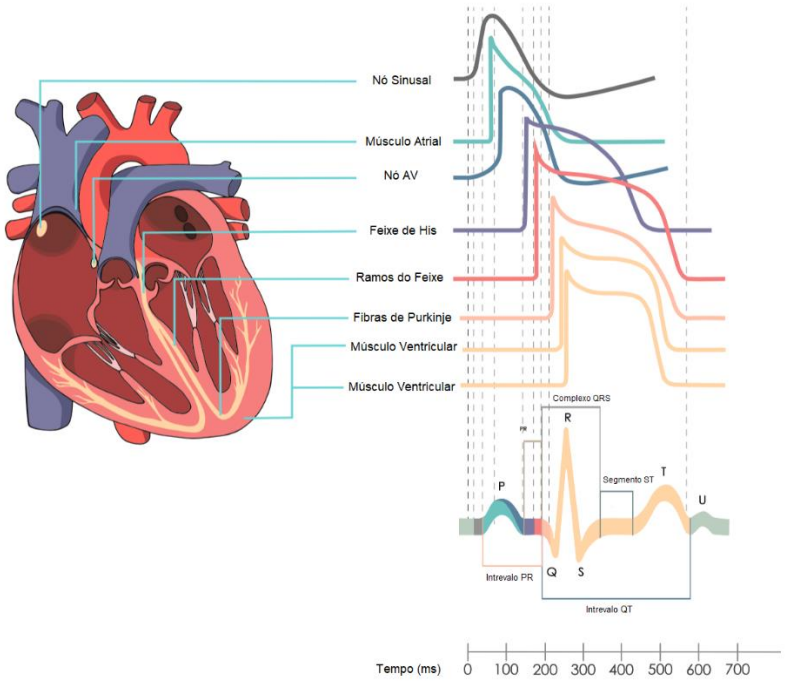

Figura 11 – Sistema de condução do coração, potenciais de ação e descrição da forma de onda do ECG (adaptado de [26]).

O ECG é amplamente utilizado na prática clínica, uma vez que fornece informações diagnósticas imediatas, sendo fundamental na seleção da terapêutica mais adequada. É também essencial na monitorização e documentação da resposta ao tratamento em diversas condições, como isquemia cardíaca, distúrbios de condução e alterações eletrolíticas [30]. Para além do seu valor clínico, o ECG tem vindo a ganhar uma maior relevância em aplicações biométricas, devido à unicidade dos sinais de cada indivíduo, à sua resistência a falsificações e à possibilidade de monitorização contínua do utilizador [29], [31].

No contexto médico, são utilizadas principalmente duas configurações padrão para adquirir um ECG, a configuração do plano frontal e a do plano horizontal. O plano frontal é analisado através de seis derivações, das quais três são bipolares (derivações I, II e III) e três são unipolares aumentadas (aVR, aVL e aVF), conforme representado na Figura 12. Estas derivam-se de elétrodos posicionados nos membros e permitem visualizar a atividade elétrica do coração num eixo vertical, como se observado de frente. Por sua vez, o plano horizontal é representado pelas seis derivações precordiais unipolares (V1 a V6), ilustradas na Figura 13, obtidas a partir de elétrodos colocados no tórax, que oferecem uma vista transversal do coração.

Embora estas configurações sejam altamente eficazes no contexto clínico, a sua aplicação em sistemas biométricos apresenta algumas limitações, nomeadamente o desconforto para o utilizador, restrições à mobilidade e a exigência de períodos prolongados de aquisição do sinal [30], [32].

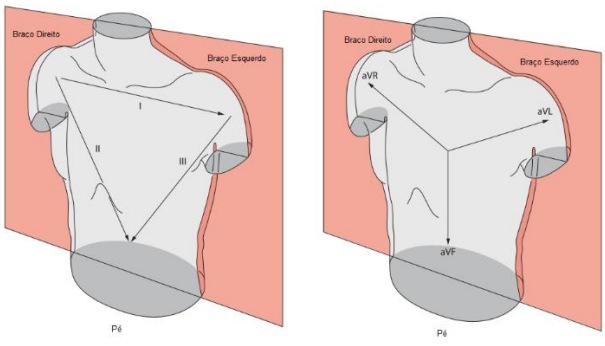

Figura 12 – Configuração dos elétrodos do plano frontal (extraído de [30]).

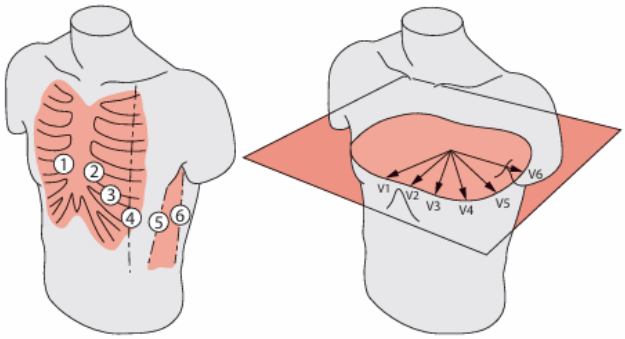

Figura 13 – Configuração dos elétrodos do plano horizontal (extraído de [30]).

2.3.1 Reconhecimento Biométrico por ECG

A autenticação ou identificação por ECG é uma tecnologia inovadora que utiliza o sinal elétrico do coração para identificar o utilizador. Em aproximadamente 10 segundos, o ECG é registado, e as características únicas das ondas geradas, como os picos (Figura 14), são analisadas para autenticar o acesso a sistemas diversos.

Este método apresenta vantagens significativas em comparação com os métodos tradicionais. Entre elas, destaca-se a maior dificuldade em ser falsificado, pois requer a utilização de hardware específico para a captura do sinal. Além disso, oferece uma garantia adicional de segurança, porque ao contrário de outros dados biométricos a existência do ECG indica a presença de uma prova de vida, assegurando que o utilizador está efetivamente presente [4], [5], [32].

A recolha de um ECG pode ser classificada em duas categorias: interna e externa. A recolha interna exige maior colaboração do utilizador, envolvendo a utilização de gel ou outros meios condutores, tornando assim o método mais intrusivo. Por outro lado, a recolha externa é mais prática e menos invasiva, embora esteja mais suscetível a ruídos, comprometendo a qualidade do sinal em comparação com a primeira abordagem [29]. Após a aquisição, o sinal de ECG é submetido a um conjunto de etapas de processamento, conforme ilustrado na Figura 15.

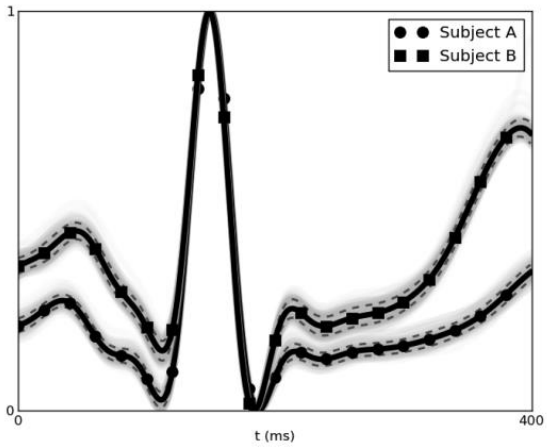

Figura 14 – Ondas de ECG de dois utilizadores diferentes (extraído de [33]).

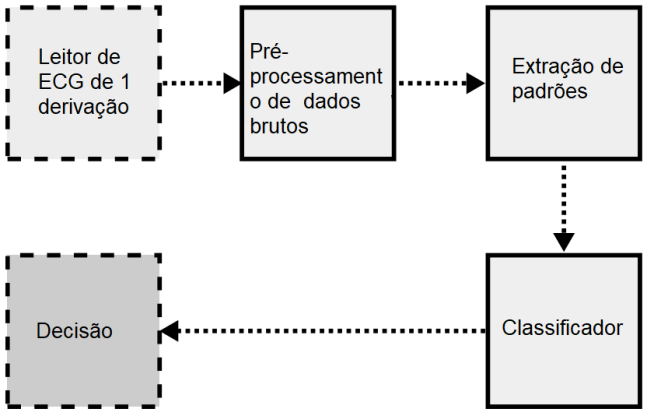

Figura 15 – Etapas do processamento do reconhecimento por ECG (adaptado de [34]).

2.3.1.1 Aquisição de ECG

A aquisição externa de um ECG pode ser realizada por diversos métodos, cada um com as suas vantagens e limitações. No âmbito da medicina, a aquisição de um ECG é realizada através das duas configurações padrão anteriormente referidas (a configuração do plano frontal e a do plano horizontal).

Embora estas configurações sejam altamente eficazes no contexto clínico, a sua aplicação em sistemas biométricos apresenta algumas limitações, nomeadamente

o desconforto para o utilizador, restrições à mobilidade e a exigência de períodos prolongados de aquisição do sinal [30].

Para contornar estes problemas, foram desenvolvidos dispositivos *Holter*, visível na Figura 16, que permitem a monitorização contínua do ECG ao longo de várias horas enquanto o utilizador realiza as suas atividades diárias. Embora tragam melhorias significativas em relação às configurações médicas tradicionais, ainda apresentam limitações em termos de conforto e viabilidade, uma vez que são desconfortáveis devido ao grande número de fios e à utilização de elétrodos húmidos que podem causar irritações na pele [32], [35].

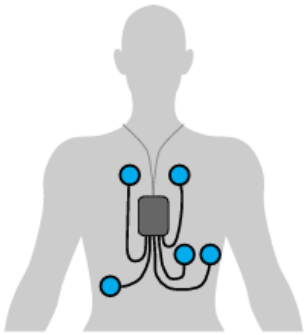

Figura 16 – Dispositivo Holter (extraído de [32]).

Para aumentar a aceitação e utilização dos sistemas biométricos de ECG, começaram a ser investigadas configurações *off-the-person*, que diferem das abordagens convencionais por permitir a captura do sinal sem a necessidade de elétrodos fixos ao corpo. Estes sistemas são amplamente utilizados na biometria, mas ainda exigem contacto físico, geralmente entre polegares ou palmas das mãos e elétrodos metálicos, o que impede uma utilização completamente livre de restrições. Apesar das suas limitações, como a maior impedância dos elétrodos secos em comparação à dos elétrodos tradicionais e a proximidade entre os elétrodos, que pode provocar variações na amplitude do sinal. Este procedimento consegue atingir sinais com uma forte semelhança em termos de morfologia em comparação com ECGs obtidos clinicamente [32], [36].

Mais recentemente, tem ocorrido um aprimoramento nas configurações *off-the-person*, tornando a biometria por ECG mais viável para aplicações comerciais. O desenvolvimento de *wearables,* representados na Figura 17, e a integração de sensores em objetos do dia a dia representam avanços importantes nesse sentido,

aproximando esta tecnologia de um uso mais prático e acessível em relação aos tradicionais dispositivos *Holter* [32], [35].

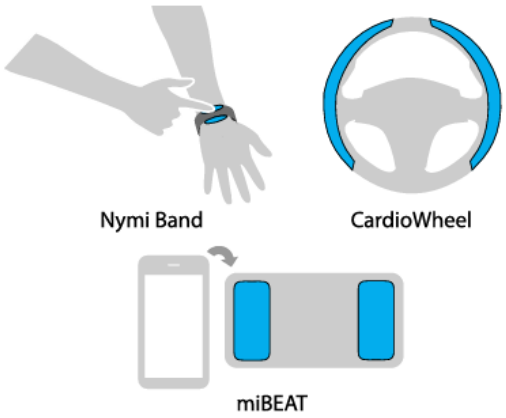

Figura 17 – Dispositivos Wearables (extraído de [32]).

2.3.1.2 Preparação do Sinal

2.3.1.2.1 Filtragem

O processo de filtragem de sinais é essencial no processamento de dados biomédicos, como no ECG, pois remove ruídos e interferências indesejadas, deste modo preservando a informação útil proveniente destes sinais. Quando o sinal é adquirido, este fica sujeito a diversos tipos de ruídos como a Interferência da Linha de Energia (PLI) causada pela rede elétrica (50/60Hz), artefactos do movimento, ruídos musculares entre outros [37].

Atualmente existem diversos métodos de filtragem para eliminar os ruídos do sinal proveniente do ECG [28], [31].

Os filtros Notch atenuam seletivamente uma faixa estreita de frequências preservando o restante sinal. São amplamente utilizados para remover interferências de frequência especifica como PLI apesar de também poderem ser utilizados noutro tipo de ruídos, garantindo que as restantes componentes do sinal se mantenham inalteradas e de fácil visualização, como ilustrado na Figura 18. Dependendo da aplicação, estes filtros podem ser implementados tanto em circuitos analógicos, quando o sinal ainda está no domínio contínuo, quanto em processamento digital, caso o sinal já tenha sido convertido para formato digital [28], [37], [38].

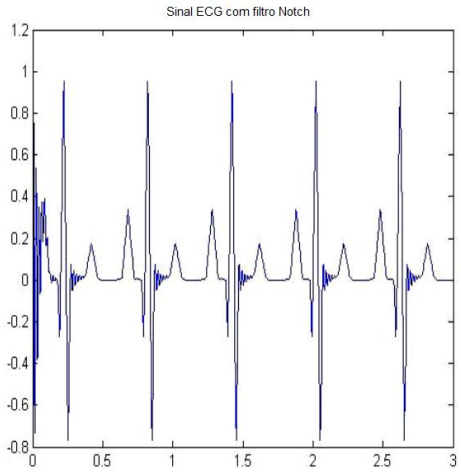

Figura 18 – Aplicação de um filtro Notch num ECG (extraído de [38]).

2.3.1.2.2 Segmentação

A segmentação do sinal ECG consiste na identificação e extração de batimentos cardíacos individuais a partir do traçado contínuo registado durante a aquisição. Esta etapa é fundamental para assegurar que os dados analisados correspondem a unidades consistentes e comparáveis entre diferentes instantes e indivíduos. No contexto da biometria baseada em ECG, a segmentação do sinal pode ser realizada através de dois métodos principais: segmentação por janelas fixas e segmentação centrada nos picos R [34], [39].

A primeira abordagem consiste na extração de segmentos do sinal com duração constante, independentemente da localização dos complexos cardíacos. Este método é simples de implementar e não depende de algoritmos de deteção de picos, sendo útil em cenários onde o sinal apresenta ruído significativo. No entanto, pode cortar os complexos P, QRS e T em posições inconsistentes, o que reduz a precisão da análise [39].

O método mais comum baseia-se na deteção dos picos R que representam os pontos de maior amplitude no complexo QRS e funcionam como referência temporal para delimitar cada batimento. A partir desses picos, extrai-se uma janela temporal fixa, num determinado intervalo de tempo, que inclui as principais componentes do ciclo cardíaco (complexos P, QRS e T), este procedimento pode ser observado na Figura 19 [34], [40].

No contexto da segmentação centrada nos picos R algumas abordagens, calculam ainda a onda média de múltiplos batimentos consecutivos, alinhados pelo pico R, de forma a reduzir o impacto do ruído e das variações locais. Para garantir segmentações fiáveis, sobretudo em condições não controladas, são utilizados algoritmos robustos de deteção de picos R, como o método de *Christov* com limiar adaptativo ou variantes do algoritmo "*multiplication of the backward distance*". Esta operação é, assim, um passo central no pipeline de processamento, influenciando diretamente a qualidade das etapas subsequentes de extração de características e classificação [31], [39].

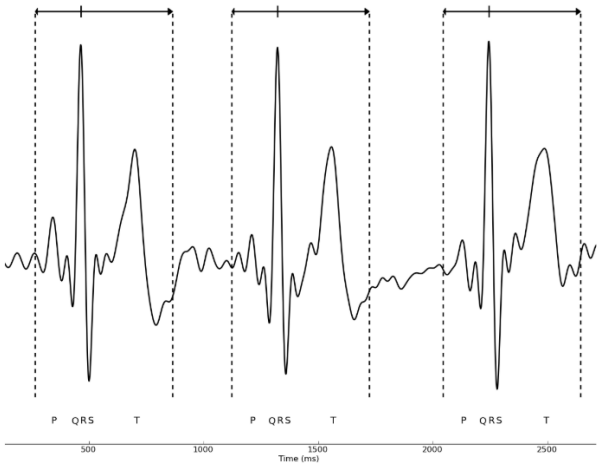

Figura 19 – Exemplo de Segmentação de um sinal ECG com picos R (extraído de [40]).

2.3.1.2.3 Extração de Características

Após a etapa de segmentação torna-se então possível a extração de características. O esquema ilustrado na Figura 20 mostra as duas principais abordagens que têm sido amplamente utilizadas na literatura: os métodos fiduciais e os métodos não fiduciais. A escolha entre estas duas abordagens influencia diretamente a complexidade do sistema, a sua robustez a ruído, e a capacidade de generalização em contextos reais [33], [39], [41].

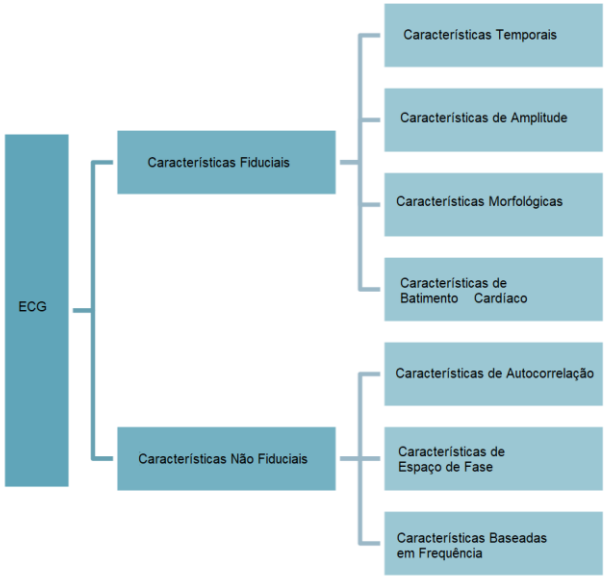

Figura 20 – Técnicas de autenticação de ECG baseadas nas caraterísticas extraídas (adaptado de [41]).

Métodos Fiduciais

Os métodos fiduciais baseiam-se na identificação de pontos anatómicos de referência no ciclo cardíaco, como os picos das ondas P, R e T, bem como os limites dos intervalos PR, QRS e QT. Estes pontos, denominados fiduciais, servem de base para a extração de características temporais e morfológicas, como durações, amplitudes e razões entre segmentos do traçado ECG. Esta abordagem requer, necessariamente, uma segmentação precisa do sinal, o que implica o uso de algoritmos de deteção de picos e filtragem robustos. A precisão na deteção dos pontos fiduciais é crítica, uma vez que qualquer erro na sua localização pode comprometer significativamente a qualidade das características extraídas e, consequentemente, o desempenho do sistema biométrico [39], [41], [42].

Métodos Não Fiduciais

Por outro lado, os métodos não fiduciais propõem uma alternativa que evita a dependência de segmentação explícita ou de deteção exata de pontos

de interesse. Em vez disso, baseiam-se em análises globais ou estatísticas do sinal, recorrendo frequentemente a técnicas de transformadas (como a transformada de *wavelet* ou de cosseno discreta), autocorrelação, entropia, ou medidas baseadas em compressibilidade e distância entre padrões. Estas abordagens apresentam como principal vantagem a maior robustez a ruído e variações intra-sujeito, uma vez que não dependem de pontos específicos no traçado, sendo particularmente adequadas para ambientes com ruído ou para sinais recolhidos em ambientes de aquisição não controlados [39], [41], [43], [44].

2.3.1.3 Classificação

Após todas as etapas de processamento e extração de características, segue-se uma das fases mais cruciais do sistema de reconhecimento biométrico, sendo a aplicação dos algoritmos de classificação. Estes algoritmos são responsáveis por interpretar as informações extraídas do sinal, atribuindo-as a uma identidade específica, sendo mais utilizados em tarefas de identificação. Existem três principais abordagens de classificação, que variam em complexidade, precisão e desempenho [32].

K-Vizinhos Mais Próximos (k-NN)

Esta abordagem, baseada em medidas de distância, é uma das mais simples e amplamente utilizadas na implementação de sistemas de biometria por ECG. O seu funcionamento consiste em comparar o novo sinal de ECG com os dados previamente armazenados na base de dados. Para isso, o vetor de características extraído do novo sinal é comparado com todos os *templates* registados, recorrendo geralmente a uma métrica como a distância Euclidiana, similaridade de cosseno, entre outras. Após o cálculo das distâncias, o sistema ordena os resultados e seleciona os *k* vizinhos mais próximos, ou seja, os *templates* mais semelhantes ao sinal apresentado. A identidade atribuída ao novo sinal será a mais frequente entre esses *k* vizinhos. No caso particular em que $k = 1$, configuração bastante comum neste tipo de sistemas, a decisão baseia-se apenas no *template* mais próximo, sendo a identidade do utilizador correspondente diretamente assumida como resultado da classificação [34], [31], [40], [45].

Máquinas de Vetores de Suporte (SVM)

As SVM são uma alternativa muito utilizada aos vizinhos mais próximos uma vez que apresentam maior desempenho para problemas com maior variabilidade ou elevado número de classes. São metodologias de classificação capazes de realizar classificações não lineares através do chamado *kernel trick,* o qual permite mapear os dados de entrada para espaços de dimensão superior, onde se torna possível encontrar separações mais eficazes entre classes. Esta técnica é particularmente útil em contextos de biometria, onde a variabilidade intra e interindivíduo pode causar sobreposição entre classes em espaços de menor dimensão [34], [32], [46].

Redes Neuronais (NN, CNN, LSTM, etc.)

Outro método bastante promissor são as redes neuronais. A principal vantagem desta abordagem reside na sua capacidade de realizar extração automática de características, eliminando a necessidade de processos manuais de *feature engineering*. Ao contrário das abordagens clássicas, que exigem a definição explícita de atributos relevantes, as redes neuronais conseguem aprender padrões úteis diretamente a partir dos sinais brutos ou minimamente processados. Estas redes são compostas por nós (ou neurónios) organizados em camadas, onde a primeira camada recebe os vetores de características e, através de funções de ativação e ligações ponderadas entre os nós, a informação é processada e conduzida até à camada final, que fornece a decisão de classificação. Esta arquitetura torna as redes neuronais particularmente eficazes na análise de grandes volumes de dados, aumentando a escalabilidade e o desempenho em sistemas com múltiplos utilizadores ou elevada variabilidade dos sinais [32], [46].

A escolha do algoritmo de classificação mais adequado depende de múltiplos fatores, como o tipo de características extraídas (fiduciais ou não fiduciais), o número de indivíduos a reconhecer, os requisitos de tempo real e a robustez esperada perante ruído e variabilidade intra-sujeito. Assim, o estudo e a otimização dos algoritmos de classificação são fundamentais para o sucesso de qualquer sistema biométrico baseado em ECG.

3. Estado de Arte

3.1 Impressão Digital

Dentro dos métodos de reconhecimento de impressão digital anteriormente referidos foi implementado um método de *minutiae-based.* Este destaca-se pela sua robustez e precisão, sendo amplamente adotado em aplicações práticas de segurança e identificação.

As aplicações mais recentes, como as propostas de Padkan et al. [47], Bakheet et al. [48] e Krivokuća e Abdulla [49], continuam a seguir abordagens de correspondência baseadas em alinhamento geométrico, aplicando transformações de rotação e translação para alinhar as minúcias entre duas impressões digitais. Estas abordagens incorporam técnicas como a construção de polígonos convexos concêntricos, algoritmos SIFT aprimorados ou histogramas de orientação das minúcias, mantendo a robustez à rotação e translação que caracteriza os métodos clássicos. Embora apresentem melhorias em robustez e eficiência, muitos destes métodos ainda exigem etapas explícitas de alinhamento e permanecem sensíveis a distorções elásticas ou impressões parciais, o que pode comprometer o desempenho em ambientes reais.

A proposta de Bahaa-Eldin [50] introduz um sistema de correspondência “leve” e eficiente para impressões de resolução inferior, adequado para sensores comerciais de baixo custo. O algoritmo evita a binarização, utilizando uma técnica de afinamento direto em tons de cinza, que melhora a extração de minúcias reais e reduz falsos positivos. A grande inovação está na representação das impressões através de vetores de contagem de minúcias, organizados em trilhas concêntricas em torno de um ponto central de cada impressão. A comparação entre impressões é feita através da diferença absoluta entre vetores e cálculo da média geométrica para verificar a correspondência, tornando-o ideal para aplicações em sistemas integrados com recursos limitados.

As principais abordagens mencionadas anteriormente encontram-se resumidas na Tabela 2, que apresenta uma comparação entre as diferentes metodologias, bases de dados e métricas de desempenho reportadas na literatura.

O método adotado nesta dissertação baseia-se na biblioteca desenvolvida por Cuevas [51] para a extração de minúcias das impressões digitais. No processo de comparação, recorre-se à abordagem proposta por Bahaa-Eldin, que, embora seja referenciada por Cuevas não é aplicada relativamente ao seu método de comparação.

Tabela 2 – Principais abordagens de reconhecimento de impressões digitais baseadas em minúcias.

Método / Autor	Abordagem Principal	Base de Dados	Métrica	Resultado
Padkan et al. (2021)	Geométrica (minúcias + polígonos)	FVC2002	EER	23% Precisão ≈ 85%
Bakheet et al. (2022)	Geométrica (minúcias + SIFT)	FVC2004	EER	2.01% Precisão ≈ 98%
Krivokuća & Abdulla (2012)	Geométrica (pré-alinhamento)	FVC2002	Precisão de alinhamento	82% em ≤ 5°; >90% em ≤ 10°
Bahaa-Eldin (2013)	Vetores concêntricos	FVC2000 / 2004	EER	1.8% – 2.0% Precisão ≈ 98%

3.2 Eletrocardiograma

3.2.1 Extração de Características

As abordagens propostas para a utilização do ECG como um dado biométrico inserem-se em duas grandes categorias no que diz respeito à extração de características: fiduciais e não fiduciais, tal como referido anteriormente.

Entre as abordagens fiduciais, destaca-se o trabalho pioneiro de Biel et al. [52], que demonstrou a viabilidade da identificação biométrica com base na análise de sinais de ECG. Através de técnicas de redução de dimensionalidade e seleção de características, os autores mostraram que é possível alcançar uma precisão elevada (cerca de 98%) mesmo utilizando apenas uma única derivação e um conjunto reduzido de características, evidenciando o potencial do ECG como biometria fiável e pouco intrusiva. Outra abordagem foi a de Silva et al. [53] onde foi proposto um sistema que combina múltiplas características extraídas de batimentos cardíacos com seleção de atributos e uma combinação de classificadores, atingindo uma taxa de acerto de 98.09% com a média de 10 batimentos e até 99.97% ao combinar decisões de múltiplas médias consecutivas.

Dos estudos não fiduciais Coutinho et al. [54] propuseram uma técnica baseada no método de compressão de dados, *Ziv-Merhav cross parsing*, que estima a complexidade cruzada entre sequências. O método atingiu 100% de precisão com

apenas 12 batimentos na sequência de teste e 13 ou mais batimentos no modelo de referência. Mesmo com apenas 5 batimentos de teste, a precisão rondava os 99.5%, demonstrando elevada eficácia e robustez, mesmo em condições com diferentes estados emocionais. Esta abordagem mostra-se promissora por dispensar etapas complexas de extração de características e por ser mais tolerante a variações morfológicas e ruído, podendo ser facilmente integrada em sistemas biométricos contínuos ou multimodais. Outro estudo com o mesmo tipo de abordagem, não fiducial, é o de Chan et al. [55], que aplicaram a transformada *wavelet* a segmentos do ECG previamente alinhados pelo pico R. Embora esse pico seja utilizado serve apenas para sincronizar os batimentos, a extração de características baseia-se nos coeficientes da *wavelet*, o que permite captar informação multiescala e reduz a sensibilidade a variações morfológicas de curta duração. Os resultados demonstraram que a métrica proposta (WDIST) alcançou uma taxa de acerto de 95% com apenas um segmento, superando significativamente os métodos tradicionais baseados em correlação ou diferença residual.

Em síntese, as abordagens fiduciais apresentam um desempenho elevado em ambientes controlados e com sinais de boa qualidade, mas tendem a ser mais sensíveis a ruído e artefactos. As técnicas não fiduciais, por sua vez, oferecem maior robustez e invariância a variações intra e interindividuais, tornando-se particularmente promissoras para aplicações práticas em contexto real. A escolha entre estas abordagens deve considerar os requisitos do sistema, a qualidade esperada dos sinais e o custo computacional associado.

3.2.2 Classificação

Com base nos artigos analisados é possível ainda obter uma visão abrangente sobre os tipos de classificação utilizados nos sistemas biométricos baseados em ECG, distinguindo tanto os métodos implementados como os contextos em que são aplicados.

De forma geral, os sistemas de classificação utilizados nestes trabalhos enquadram-se maioritariamente em abordagens supervisionadas, onde os padrões extraídos dos sinais ECG são comparados com modelos previamente armazenados durante a fase de registo. Em particular, destaca-se o uso recorrente do classificador *k*-NN, como em Canento et al. [40] e Lourenço et al. [34] , onde a decisão é baseada na proximidade (geralmente em termos de distância Euclidiana ou cosseno) entre o padrão

de teste e os padrões armazenados. Esta abordagem é especialmente útil quando os vetores de características apresentam uma estrutura relativamente simples, como é o caso de segmentos normalizados de batimentos cardíacos.

Existem ainda outros métodos de classificação mais complexos como a utilização de redes neuronais ou SVM, particularmente em abordagens não fiduciais, onde as características extraídas não se baseiam em pontos anatómicos específicos do sinal (como picos R), mas sim em janelas do sinal bruto ou em representações transformadas. Nestes casos, a robustez à variabilidade inter e intraindivíduos é crucial, e métodos de classificação com maior capacidade de generalização mostram-se mais eficazes. Um exemplo representativo deste tipo de métodos é o trabalho de Hejazi et al. [43], que propuseram um sistema de autenticação baseado em ECG usando uma abordagem não fiducial que combina autocorrelaçao com métodos de redução de dimensionalidade. A classificação é feita via SVM com um *kernel gaussiano*. O sistema demonstrou elevada robustez com dados não clínicos e curtos segmentos de ECG, atingindo taxas de reconhecimento de até 88.4% por janela e 94.5% por sujeito em validação cruzada.

É importante referir que o tipo de classificação utilizado depende do método de extração de características, nomeadamente se este segue uma abordagem fiducial ou não fiducial. Enquanto classificadores simples como o *k*-NN são atrativos pela sua simplicidade e interpretabilidade, abordagens mais complexas oferecem melhor desempenho em cenários com maior ruído ou variabilidade fisiológica, ainda que à custa de maior custo computacional e necessidade de treino.

Assim, observa-se uma diversidade de métodos de classificação aplicados ao reconhecimento biométrico por ECG, sendo a escolha do classificador um elemento crítico que deve ser feito em função do tipo de sinal, das características extraídas e dos requisitos específicos da aplicação. Os métodos referidos na literatura encontram-se resumidos na Tabela 3.

Tabela 3 – Comparação das principais abordagens de reconhecimento biométrico por ECG quanto ao tipo de abordagem, extração de características e desempenho.

Autor / Ano	Tipo de Abordagem	Extração de Características	Resultados Principais
Biel et al. (2001)	Fiducial	Redução de dimensionalidade em pontos fiduciais	Precisão ≈ 98%
Silva et al. (2014)	Fiducial	Múltiplas *features* por batimento + seleção de atributos	98.09% (10 batimentos); 99.97% (múltiplas médias)
Coutinho et al. (2013)	Não fiducial	*Ziv-Merhav cross parsing*	100% (≥12 batimentos); 99.5% (5 batimentos)
Chan et al. (2008)	Não fiducial	Transformada *Wavelet*	Precisão ≈ 95% (1 segmento)
Canento et al. (2012)	Fiducial	Segmentos normalizados de batimentos	Resultados robustos em reconhecimento individual
Lourenço et al. (2012)	Fiducial	Segmentos normalizados de batimentos	Autenticação fiável em tempo real
Hejazi et al. (2017)	Não fiducial	Autocorrelação + redução de dimensionalidade	88.4% por janela; 94.5% por sujeito (CV)

3.3 Sistemas Bimodais

A Figura 21 apresenta uma classificação geral dos sistemas de autenticação baseados em ECG, distinguindo entre abordagens unimodais, que utilizam apenas características extraídas do próprio sinal de ECG, e abordagens multimodais, que combinam o ECG com outras fontes de informação, como outras biometrias, *passwords* ou técnicas de criptografia.

Neste contexto, os sistemas biométricos multimodais, que combinam o reconhecimento de impressões digitais e sinais provenientes ECG estão a ganhar uma elevada atenção devido à sua maior segurança contra-ataques de falsificação e *Presentation Attacks*. As impressões digitais fornecem um identificador fisiológico

confiável, enquanto os sinais de ECG garantem a deteção de prova de vida, abordando vulnerabilidades dos sistemas unimodais [2], [3], [8].

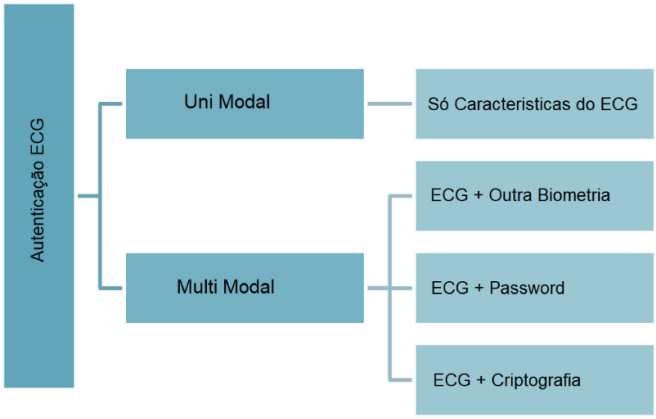

Figura 21 – Técnicas de autenticação de ECG baseadas na modalidade utilizada (adaptado de [41]).

Os métodos de fusão em sistemas biométricos multimodais são geralmente classificados em dois grandes grupos: fusão antes da correspondência e fusão após correspondência. A fusão prévia à correspondência inclui a fusão ao nível do sensor, na qual os sinais brutos de diferentes modalidades são combinados diretamente; e a fusão ao nível das características, na qual os vetores de características extraídos individualmente de cada biometria são integrados num vetor composto, que é então utilizado como entrada para o processo de classificação. Este último método pode oferecer maior precisão, mas exige que os dados originados sejam compatíveis em dimensão e escala. Já a fusão após a correspondência é subdividida em três categorias principais: a fusão ao nível de *score*, que combina os resultados de similaridade obtidos individualmente por cada modalidade através de regras como soma ponderada ou média; a fusão ao nível de decisão, que integra decisões binárias (aceitar ou rejeitar) provenientes de cada subsistema biométrico com base em regras lógicas, como maioria ou conjunção; e a fusão ao nível de ranking, que utiliza a ordenação dos candidatos por cada modalidade e funde esses rankings para determinar o resultado final [56], [57], [58], [59].

O estudo de Arteaga-Falconi et al. [5] propõe um sistema bimodal que combina impressão digital e ECG com fusão ao nível da tomada de decisão. A impressão digital é processada através da extração e correspondência de minúcias utilizando as ferramentas do NBIS, enquanto o ECG é analisado com recurso a um classificador SVM

com um *kernel* radial. O sistema permite realizar a autenticação em apenas quatro segundos de sinal ECG, reduzindo assim o tempo necessário para aquisição. Os resultados obtidos demonstraram uma Taxa de Erro Igual (EER) de 0.46%, superando tanto a impressão digital utilizada de forma isolada, que apresentou um EER de 1.18%, como o próprio ECG analisado de forma isolada. Os autores destacam ainda a aplicabilidade deste sistema em contextos como aeroportos e fronteiras, onde seria possível autenticar viajantes colocando os dois polegares num único dispositivo capaz de captar simultaneamente a impressão digital e o sinal cardíaco.

Por sua vez, Manjunathswamy et al. [2] desenvolveu uma arquitetura multimodal que também combina ECG e impressão digital, mas com uma abordagem de fusão a nível de características. Neste caso, foram extraídas doze características do ECG e duas da impressão digital, sendo a decisão final baseada na validação de pelo menos 75% dessas características, ou seja, onze em catorze. O sistema foi testado com uma base de dados própria e obteve resultados bastante robustos, com uma Taxa de Falsa Rejeição (FRR) de 0% e uma Taxa de Falsa Aceitação (FAR) de apenas 2.5%. Além disso, a fusão das duas modalidades biométricas permitiu alcançar uma especificidade de 100% e uma exatidão de 97.5%, superando claramente o desempenho de cada biometria utilizada de forma isolada.

Entre as abordagens encontradas na literatura, destaca-se o trabalho de Komeili et al. [4] cuja proposta serviu de base para a implementação realizada nesta dissertação. Os autores apresentam um sistema multimodal que combina ECG e impressão digital com o objetivo de realizar não só a autenticação do utilizador, mas também a deteção de prova de vida. A fusão é efetuada ao nível do *score*, combinando os resultados dos módulos de reconhecimento e vitalidade através de regras como soma ponderada. O ECG é adquirido a partir das pontas dos dedos, o que permite a integração física com o sensor de impressão digital. A abordagem inclui ainda um mecanismo de atualização automática de *templates*, que permite adaptar o sistema a variações fisiológicas do utilizador ao longo do tempo. Os resultados obtidos demonstraram uma melhoria significativa na robustez e na capacidade de detetar ataques de *spoofing*, com uma Taxa de Erro Igual (EER) de apenas 2.6% utilizando 30 segundos de ECG. Este trabalho evidencia o potencial da fusão de biométricas fisiológicas e comportamentais para aplicações em cenários de segurança reforçada.

As principais abordagens multimodais que combinam ECG e impressão digital encontram-se resumidas na Tabela 4, que apresenta uma comparação entre as estratégias de fusão, metodologias e resultados reportados na literatura.

Os estudos analisados anteriormente demonstram que a integração de ECG e impressão digital oferece vantagens significativas na segurança e robustez dos sistemas biométricos. A possibilidade de deteção de prova de vida proporcionada pelo ECG, como explorado por Komeili et al., torna o sistema mais resistente a ataques de *spoofing*, eliminando a principal vulnerabilidade da impressão digital. Apesar disso, continua a haver desafios relevantes, como o custo e a complexidade dos sensores de ECG, o tempo de aquisição do sinal e a escassez de conjuntos de dados multimodais.

A escolha da abordagem de fusão mais adequada depende da estrutura do sistema, da compatibilidade entre as modalidades utilizadas e dos requisitos de desempenho pretendidos. Nesse contexto, a combinação de uma biometria consolidada, como a impressão digital, com uma biometria fisiológica como o ECG, revela-se especialmente promissora, permitindo verificar não apenas a identidade, mas também a presença de vida do utilizador. Superar as atuais barreiras técnicas e práticas será essencial para desbloquear todo o potencial dos sistemas multimodais, sobretudo em aplicações críticas como segurança bancária, militar ou controlo de fronteiras, onde a fiabilidade da autenticação é crucial [2], [4], [5].

Tabela 4 – Principais sistemas biométricos multimodais baseados em ECG e impressão digital.

Autor / Ano	**Nível de Fusão**	**Abordagem**	**Métricas**	**Resultados Principais**
Arteaga-Falconi et al. (2016)	Decisão	Combina minúcias de impressão digital (NBIS) com ECG processado via SVM	EER	EER = 0.46%; superou o desempenho isolado (ECG: 1.18%)
Manjunathswamy et al. (2019)	Características	Fusão de 12 features de ECG + 2 da impressão digital	FRR, FAR, Especificidade, Exatidão	FRR = 0%; FAR = 2.5%; Especificidade = 100%; Exatidão = 97.5%
Komeili et al. (2018)	Score	Combina módulos de reconhecimento e prova de vida via soma ponderada	EER	EER = 2.6%; elevada robustez e deteção de spoofing

4. Materiais e Métodos

Este capítulo descreve os materiais e metodologias utilizadas no desenvolvimento do sistema. O objetivo é apresentar, de forma estruturada, os dispositivos, bibliotecas e algoritmos utilizados, desde a aquisição de dados até à fase de fusão de resultados.

A Figura 22 representa a arquitetura do sistema biométrico bimodal desenvolvido, que combina impressão digital e ECG para identificação e autenticação. Já a Figura 23 ilustra a arquitetura do *harware* implementado. O fluxo abrange desde a aquisição dos sinais até à fusão dos resultados sendo que os detalhes de cada componente são descritos nas secções seguintes.

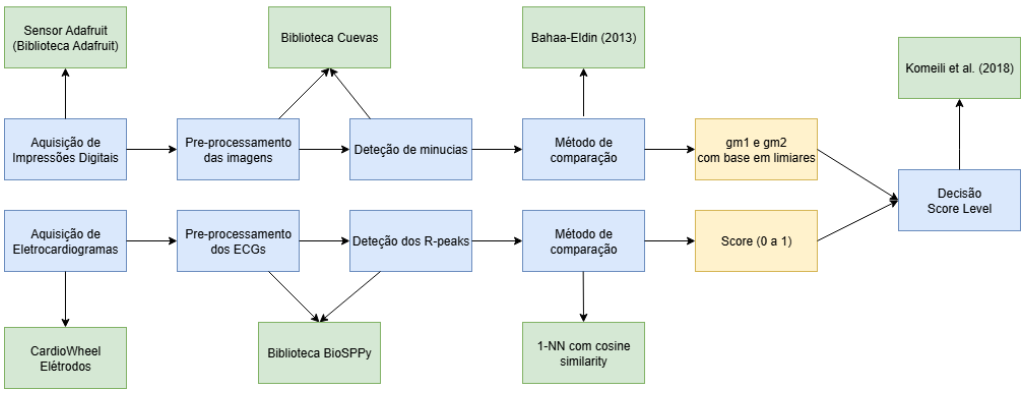

Figura 22 – Arquitetura do sistema bimodal implementado.

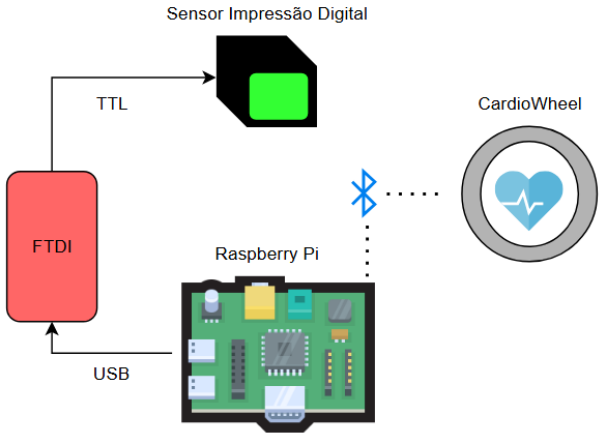

Figura 23 – Arquitetura do hardware implementado.

4.1 Impressão Digital

A implementação do sistema de reconhecimento de impressões digitais baseou-se na utilização de um sensor ótico da Adafruit Industries [60], conforme representado na Figura 24, amplamente utilizado em aplicações biométricas devido à sua fiabilidade na captura de imagens. Para a aquisição das impressões digitais, recorreu-se à biblioteca oficial fornecida pela empresa [61], que facilita a comunicação com o sensor e disponibiliza funções de captura e armazenamento de imagens. A ligação ao Raspberry Pi 5 [62] foi realizada através de um conversor FTDI232 [63] que permite a conversão de sinais Série TTL (Tabela 5) para USB, como representado na Figura 25. Todo o software foi desenvolvido na linguagem de programação Python, devido à vasta gama de bibliotecas disponíveis para tarefas como comunicação com hardware, processamento de imagem e análise de dados biométricos.

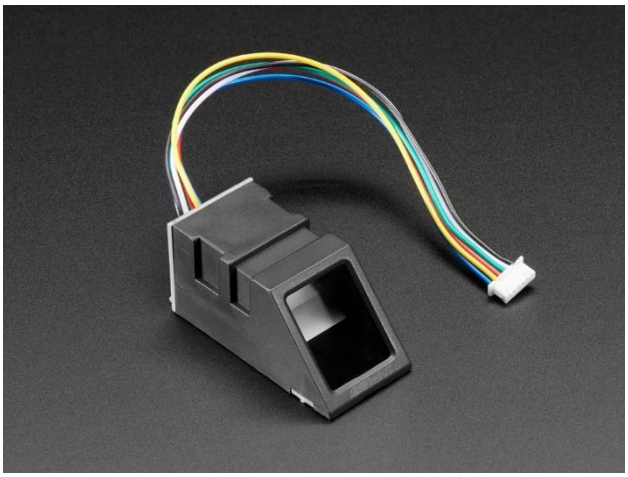

Figura 24 – Sensor ótico de impressão digital da Adafruit (extraído de [60]).

Tabela 5 – Tabela de características do sensor (adaptado de [60]).

Parâmetro	Especificação
Energia	DC 3.6V–6.0V
Interface	UART (nível lógico TTL) / USB 1.1
Corrente de funcionamento	Típica: 100 mA Pico: 150 mA
Modo de correspondência	1:1 e 1:N
Taxa de transmissão (Baud rate)	(9600*N) bps, N = 1–12 (valor por omissão N = 6)
Tamanho do ficheiro de caracteres	256 bytes
Tempo de aquisição de imagem	< 1 s
Tamanho do template	512 bytes
Capacidade de armazenamento	120 / 375 / 880
Nível de segurança	5 (1, 2, 3, 4, 5 [mais elevado])
FAR (Taxa de Aceitação Indevida)	< 0,001%
FRR (Taxa de Rejeição Indevida)	< 0,1%
Tempo médio de pesquisa	< 1 s (1:880)
Dimensão da janela	14 mm × 18 mm
Ambiente de funcionamento	Temperatura: -10 ºC a +40 ºC
Ambiente de armazenamento	Temperatura: -40 ºC a +85 ºC
Dimensões exteriores	Módulo: 42 × 25 × 8,5 mm Sensor: 56 × 20 × 21,5 mm
Dimensões exteriores	56 × 20 × 21,5 mm

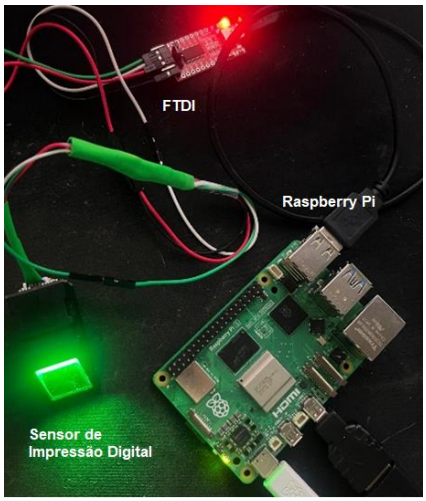

Figura 25 – Ligação do sensor ótico ao Raspberry Pi 5.

Após a captura das imagens, foi utilizada a biblioteca de Cuevas [51] para o pré-processamento das impressões digitais, dado que esta se mostrou particularmente eficaz no tratamento de imagens de menor qualidade. Neste processo, foram aplicadas várias transformações, incluindo normalização da intensidade, segmentação da área útil, geração dos mapas de orientação e frequência, aplicação de filtros de *Gabor* e posterior esqueletização das cristas, conforme ilustrado na Figura 26. Concluídas estas etapas, procedeu-se à extração de minúcias, identificando-se as terminações das cristas (com pontos vermelhos) e as bifurcações (com pontos verdes). Para lidar com a ocorrência de falsas minúcias nas bordas, foi adicionado um filtro à função de deteção, capaz de eliminar pontos fora de uma margem pré-definida e também os que se encontram demasiado próximos entre si, conforme ilustrado na Figura 27.

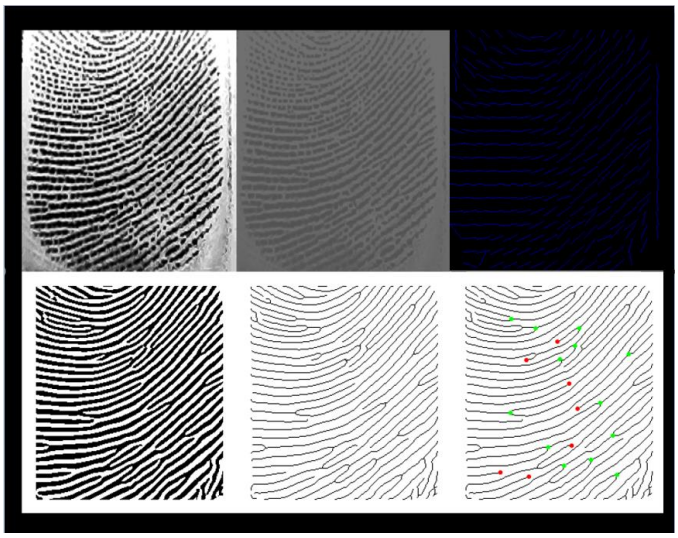

Figura 26 – Etapas de pre-processamento de imagem. a) Imagem original; b) Normalização; c) Mapeamento de cristas; d) Binarização; e) Afinamento; f) Extração de minúcias.

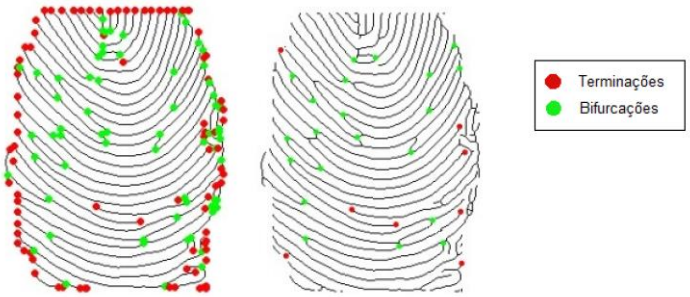

Figura 27 – Aplicação do filtro de remoção de falsas minúcias.

Para a comparação entre impressões digitais, foram considerados apenas o indicador, o dedo médio e o anelar, por apresentarem áreas com melhor definição e menor nível de ruído, facilitando a extração de minúcias. Após a seleção das melhores amostras, foi implementado o método de comparação de Bahaa-Eldin [50], conforme descrito anteriormente. O processo incluiu o cálculo do ponto médio (*core point*) e a criação de trilhas concêntricas ao seu redor. Com o objetivo de melhorar a eficácia do sistema, foram geradas 14 trilhas, espaçadas entre si por 10 pixels, conforme visualizado na Figura 28. Após a criação dos vetores de características, que representam a contagem de minúcias de terminação e bifurcação em cada trilha concêntrica ao redor do ponto central, a comparação entre duas impressões digitais foi realizada com base na distância euclidiana entre os vetores correspondentes. A partir dessa comparação, obteve-se os valores $sum1$ e $sum2$, que representam a diferença global entre as minúcias de terminação e de bifurcação, respetivamente.

Para a calibração dos limiares $sum1$ e $sum2$, foi selecionado um conjunto de três impressões digitais de referência do mesmo utilizador. Cada uma destas impressões foi comparada com todas as restantes disponíveis no conjunto de dados, incluindo impressões genuínas e impostoras. A partir dessa análise, os limiares foram ajustados de forma iterativa, de modo que apenas as comparações genuínas fossem classificadas como verdadeiras, enquanto todas as restantes resultassem em falsas.

Após várias iterações, observou-se que os valores $sum1 \leq 27$ e $sum2 \leq 18$ permitiram atingir esse equilíbrio, garantindo uma correta aceitação das impressões genuínas e rejeição das impostoras. Estes valores foram, assim, definidos como limiares finais para os testes subsequentes.

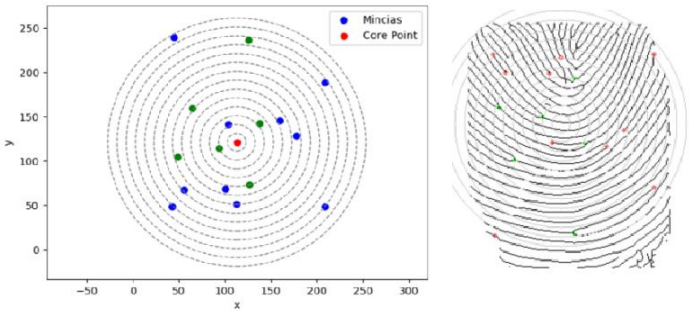

Figura 28 – Distribuição de minúcias em trilhas.

4.2 Eletrocardiograma

A implementação do sistema de reconhecimento por ECG foi realizada com recurso ao CardioWheel [6], um dispositivo desenvolvido pela CardioID, como ilustrado na Figura 29. Este equipamento consiste num volante com elétrodos integrados e com um couro condutor, capaz de adquirir sinais ECG de forma não intrusiva durante a condução. A comunicação com o Raspberry Pi é feita via Bluetooth, o que simplificou significativamente a integração do dispositivo no sistema desenvolvido.

Figura 29 – CardioWheel [6].

Após a aquisição dos sinais ECG, estes foram processados com o auxílio da biblioteca BioSPPy [64], que permite extrair automaticamente características relevantes do sinal, nomeadamente os batimentos individuais. O processamento inclui a deteção dos picos R, segmentação dos ciclos cardíacos e extração de múltiplos *templates* representativos da atividade elétrica do coração. Estes *templates* foram posteriormente normalizados em comprimento e amplitude, recorrendo à técnica de re-amostragem seguida de normalização *z-score*, garantindo assim a uniformidade entre diferentes registos. A partir do conjunto de *templates* extraídos, foi calculado um *template* médio,

que serve como vetor de características representativo do ECG de cada utilizador. Este vetor foi utilizado tanto para autenticação (1:1) como para identificação (1:N), utilizando uma métrica de distância apropriada para vetores normalizados.

A extração de características seguiu uma abordagem fiducial, uma vez que envolveu a deteção dos picos R e a sua posterior utilização para segmentar os ciclos cardíacos.

Para a classificação, foi adotado o método *k*-NN com $k = 1$, dado que, no processo de identificação, os *templates* foram comparados individualmente em vez de simultaneamente, como observado na Figura 30. A métrica utilizada para medir a semelhança entre *templates* foi a similaridade cosseno, por se adequar à comparação entre vetores normalizados. Esta métrica retorna um *score* entre 0 e 1, sendo este valor utilizado para avaliar a correspondência entre *templates* e, consequentemente, a eficácia do método.

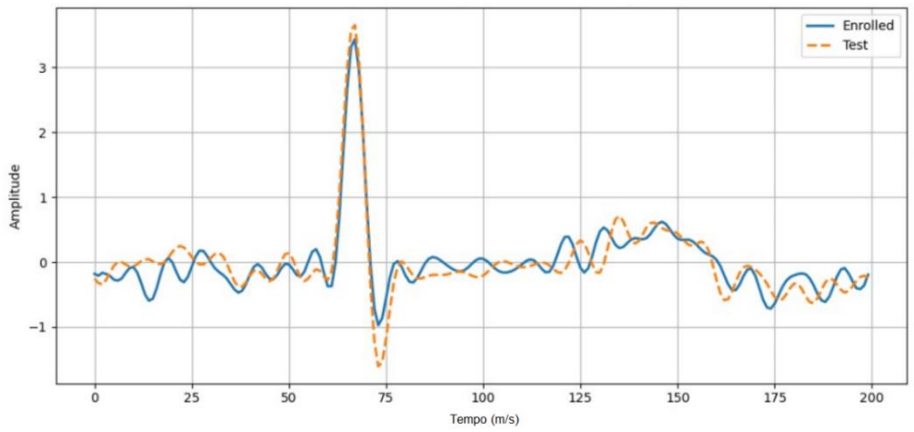

Figura 30 – Gráfico de comparação entre o template médio do ECG registado e o de teste para um sujeito ilustrativo.

4.3 Sistema Bimodal

A abordagem adotada para o sistema biométrico bimodal foi inspirada no trabalho de Komeili et al. [4], recorrendo à fusão das modalidades ao nível de *score*.

Para permitir a fusão linear, foram colocados ambos os *scores* na mesma escala. O *score* do ECG é obtido através da similaridade de cosseno, como referido anteriormente, originando assim um valor no intervalo [0, 1]. Para o *score* de impressão se encontrar na mesma escala foi necessário aplicar a Equação (1), sendo que $gm1$ e $gm2$ são constantes de calibração, que neste caso tomaram os valores dos limites anteriormente referidos para obtenção de melhores resultados.

Nesta estratégia, os *scores* obtidos individualmente a partir do ECG e da impressão digital foram combinados através da Equação (2), na qual cada modalidade contribui com um peso distinto. Devido à maior estabilidade e fiabilidade dos sinais ECG no contexto deste projeto, foi atribuído a esta modalidade um peso superior, compensando assim a variabilidade observada na leitura das impressões digitais.

$$Score\ FP = \frac{1}{1+\frac{sum1}{gm1}+\frac{sum2}{gm2}} \qquad (1)$$

$$Score\ Total = 0.7 \times Score\ ECG + 0.3 \times Score\ FP \qquad (2)$$

O código foi inteiramente implementado em Python, estruturado em torno de um menu interativo com opções para registo, identificação (1:N), autenticação (1:1) e eliminação de registos existentes, conforme observado na Figura 31.

```
===== SISTEMA MULTIMODAL (ECG + Impressao Digital) =====
1 - Registo
2 - Identificacao (1:N)
3 - Autenticacao (1:1)
4 - Limpar todos os registos
0 - Sair
Escolha uma opcao:
```

Figura 31 – Menu do sistema bimodal.

5. Resultados

Antes de iniciar a fase de avaliação, foi necessário substituir as aquisições realizadas com o sistema desenvolvido por bases de dados públicas. Durante os testes preliminares, verificou-se que a componente de impressão digital apresentava erros de correspondência em torno dos 50 %, o que indicava um comportamento praticamente aleatório. Esta limitação resultou do nível de ruído introduzido pelo sensor ótico utilizado, que comprometia a deteção fiável das minúcias e, consequentemente, a geração de *templates* consistentes. Assim, optou-se por recorrer a bases de dados de referência, garantindo condições controladas e resultados reprodutíveis que permitissem avaliar objetivamente o desempenho do sistema biométrico proposto.

Para a avaliação prática do sistema foram consideradas duas bases de dados para as modalidades distintas. Para a impressão digital foi considerada a base de dados FVC2002 [65], utilizada em abordagens anteriores, e para o ECG foi considerada a base de dados do PhysioBank [66].

5.1 Procedimento Experimental

O procedimento experimental consistiu em associar cada sujeito da base de dados de ECG a um sujeito da base de dados de impressões digitais, criando pares multimodais correspondentes. Para cada indivíduo foram considerados três registos independentes, permitindo realizar múltiplas comparações genuínas e impostoras e avaliar a consistência intrapessoal e a separabilidade interpessoal.

Durante a preparação dos *templates* de impressão digital, foram ajustados os parâmetros de extração de características de modo a otimizar a captação das minúcias. O número de trilhas concêntricas foi reduzido de 14 para 5, e a largura de cada trilha aumentada de 10 para 31 pixéis, permitindo abranger áreas mais amplas da impressão e captar melhor a distribuição das minúcias relevantes, reduzindo simultaneamente o impacto do ruído periférico. A seleção dos parâmetros foi realizada de forma empírica, após a avaliação de diferentes combinações experimentais.

Nesta fase, optou-se por avaliar apenas o cenário de autenticação (1:1) e não o de identificação (1:N). Esta decisão deveu-se a dois fatores principais: por um lado, o

número relativamente reduzido de participantes não permitiria obter métricas estatisticamente representativas para o caso de identificação; por outro, o objetivo central deste trabalho consistiu em validar a viabilidade de um sistema biométrico bimodal em contextos de autenticação, que correspondem ao caso de uso mais recorrente em aplicações práticas de segurança.

Posteriormente, foram efetuadas todas as combinações possíveis de autenticação sendo que com 3 registos por pessoa e utilizando 10 utilizadores deu um total de 30 cenários de genuíno e 810 cenários de impostor. Todos os *scores* obtidos em ambos os cenários foram registados automaticamente e exportados para um ficheiro CSV, que serviu de base à análise quantitativa dos resultados apresentada no capítulo seguinte.

5.2 Métricas Obtidas

Os resultados quantitativos obtidos a partir das comparações genuínas e impostoras são apresentados na Tabela 6, onde se encontram resumidos os valores médios das principais métricas de desempenho para cada modalidade individual e para o sistema resultante da fusão multimodal.

Tabela 6 – Resultados das métricas obtidas com 10 utilizadores.

	FAR (%)	*FRR (%)*	*EER (%)*
Impressão Digital	35.56	33.33	34.44
Eletrocardiograma	10.86	10.00	10.43
Fusão	7.16	6.67	6.91

Os testes foram realizados numa única execução determinística, na qual todas as combinações possíveis entre registos genuínos e impostores foram avaliadas de forma exaustiva. Deste modo, não foi introduzida qualquer aleatoriedade no processo experimental, e as métricas obtidas correspondem a valores fixos para o conjunto de dados considerado, não sendo aplicável o cálculo de desvio-padrão.

Com base nos resultados obtidos, ilustrados na Tabela 6, é possível observar diferenças significativas entre as modalidades distintas. O ECG apresentou um melhor desempenho, com um EER de 10.43% e valores de FAR e FRR por volta dos 10%. Este resultado demonstra uma boa consistência intrapessoal e uma clara distinção entre registos genuínos e impostores, refletindo a estabilidade temporal do sinal cardíaco e a fiabilidade do processo de extração de *templates,* embora com margem para melhorias (Figura 32).

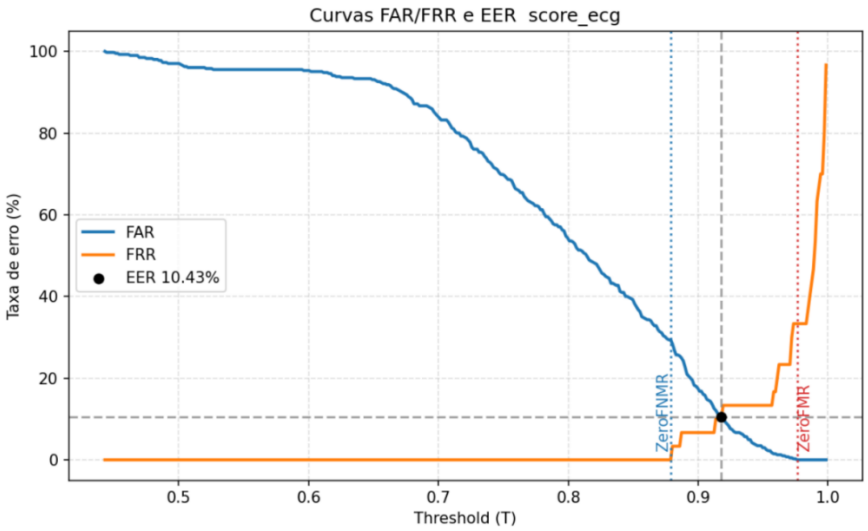

Figura 32 – Curvas FAR/FRR e EER do sistema baseado em ECG.

Ao contrário do ECG, o sistema de impressão digital apresentou um desempenho inferior, com um EER de 34.44%. As curvas FAR e FRR mostram uma interseção num ponto mais afastado do ideal, evidenciando uma maior sobreposição entre as distribuições genuína e impostora, possivelmente associada à variabilidade presente nas amostras da base de dados e à limitação do algoritmo de extração de minúcias, como observado na Figura 33.

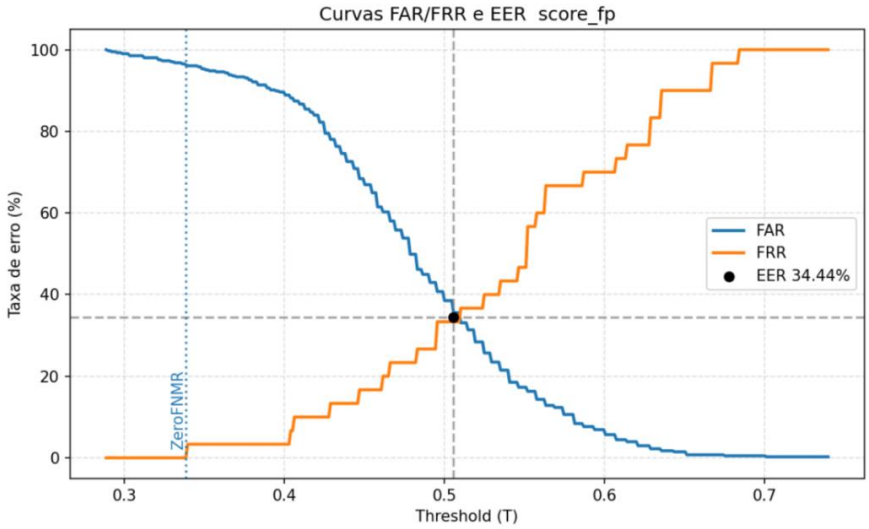

Figura 33 – Curvas FAR/FRR e EER do sistema baseado em impressão digital.

Já a fusão multimodal, obtida por combinação ponderada dos *scores* das duas modalidades, resultou num EER de apenas 6.91%, evidenciando uma melhoria significativa face às abordagens unimodais. Além de reduzir simultaneamente as taxas de FAR e FRR, a fusão proporcionou uma melhor discriminação entre utilizadores genuínos e impostores, demonstrando a eficácia da integração entre diferentes características fisiológicas (Figura 34).

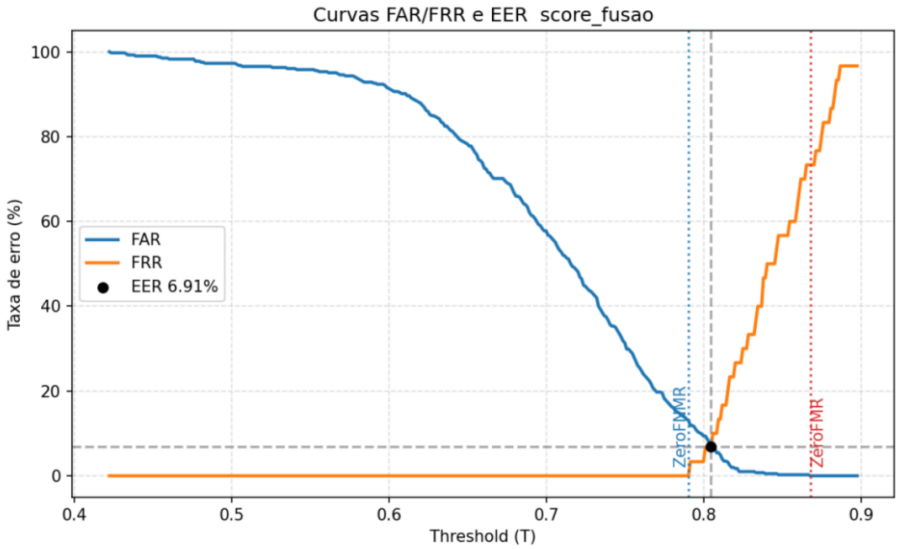

Figura 34 – Curvas FAR/FRR e EER do sistema multimodal.

Em relação à curva ROC, o sistema multimodal é o que apresenta a maior área sob a curva (AUC = 0.982), evidenciando uma elevada capacidade de distinção entre genuínos e impostores. A proximidade à fronteira ideal (canto superior esquerdo) indica um comportamento altamente robusto e generalizável, com baixa taxa de falsos positivos mesmo em limiares mais exigentes. Por contraste, observa-se que a curva correspondente à impressão digital encontra-se próxima da diagonal, o que reflete um comportamento quase aleatório e confirma o baixo poder discriminativo desta modalidade isolada (Figura 35).

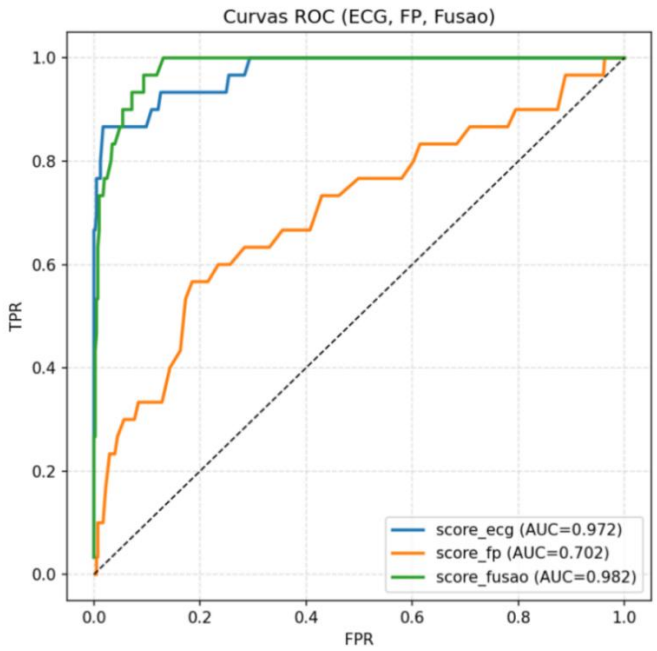

Figura 35 – Curvas ROC para ECG, impressão digital e fusão multimodal.

Apesar da boa consistência global do sistema, observou-se uma limitação relevante na componente de impressão digital, relacionada com o desalinhamento geométrico entre amostras genuínas. A Figura 36 ilustra este fenómeno, apresentando duas amostras do mesmo dedo. Nota-se uma rotação e deslocamento visíveis que afetam diretamente a correspondência das minúcias, levando a que várias terminações e bifurcações reais não coincidam entre si. Além disso, verificou-se a ocorrência de falsas minúcias nas regiões de borda, principalmente em terminações geradas por ruído ou cortes parciais nas cristas. O filtro desenvolvido para eliminar esses artefactos não foi totalmente eficaz, permitindo a permanência de algumas falsas terminações que contribuíram para o aumento da taxa de erro. Como resultado, o número de

correspondências válidas diminui, reduzindo o *score* de aceitação e elevando o EER da componente de impressão digital. Estas limitações explicam o desempenho inferior face à modalidade ECG e reforçam a necessidade de incluir uma etapa de pré-alinhamento geométrico e de filtragem de bordas mais robusta nas iterações futuras do sistema.

Figura 36 – Comparação entre duas impressões digitais do mesmo utilizador com diferentes alinhamentos.

De forma global, os resultados obtidos confirmam a viabilidade e eficácia do sistema proposto, demonstrando que a fusão de características fisiológicas e morfológicas resulta num sistema de autenticação mais equilibrado, preciso e resistente a variações e tentativas de falsificação.

5.3 Comparação com Literatura

Comparando os resultados obtidos nesta dissertação com os de outros trabalhos realizados é possível observar diferenças significativas a nível das métricas que medem o desempenho das modalidades tanto de forma individual como na fusão entre elas.

No sistema desenvolvido nesta dissertação, embora o ECG tenha apresentado resultados aceitáveis, a modalidade de impressão digital revelou limitações significativas, como referido anteriormente, refletindo assim em métricas de erro bastante superiores às do estudo de referência. As diferenças verificadas entre os resultados obtidos nesta dissertação e os de Manjunathswamy et al. [2], devem-se a sobretudo aos métodos de comparação, uma vez que, no artigo, foram aplicados Sistemas de Identificação Automática de Impressões Digitais (AFIS) [67], enquanto

neste trabalho foi aplicado um método experimental baseado em *Crossing Number* e distribuição de trilhas concêntricas.

De igual modo, ao comparar com os resultados de Arteaga-Falconi et al. [5], também se observam diferenças relevantes tanto ao nível metodológico como de desempenho. No referido estudo, a modalidade de impressão digital foi processada com algoritmos desenvolvidos pela NIST, amplamente utilizados em sistemas biométricos, e considerados soluções consolidadas e altamente otimizadas para extração e correspondência de minúcias. Para o ECG, apesar de também ter sido feita uma extração de *features* fiduciais, recorreu-se à utilização de um classificador SVM com *kernel* RBF, capaz de lidar de forma robusta com a variabilidade inter e intra-sujeito.

6. Conclusões e Trabalho Futuro

6.1 Conclusões

Nesta dissertação, alcançaram-se os objetivos definidos, centrados na implementação e validação de um sistema biométrico bimodal. O trabalho desenvolvido permitiu demonstrar a viabilidade de combinar características fisiológicas e morfológicas — nomeadamente o ECG e a impressão digital — para autenticação de utilizadores.

O sistema proposto foi integralmente desenvolvido e testado num ambiente real, recorrendo a hardware acessível e software modular, demonstrando a possibilidade de implementar soluções biométricas avançadas em plataformas de baixo custo, como o Raspberry Pi. Este contributo é particularmente relevante no contexto de dispositivos portáteis e aplicações embebidas, onde a limitação de recursos constitui um fator crítico.

Os resultados experimentais confirmaram a robustez e consistência da modalidade de ECG, evidenciando a sua fiabilidade como biometria fisiológica. A componente de impressão digital, por outro lado, apresentou limitações associadas ao pré-processamento e à segmentação das imagens, como referido anteriormente, o que reduziu a precisão da correspondência. Ainda assim, a integração multimodal demonstrou ganhos claros em termos de equilíbrio e segurança, validando o potencial da fusão biométrica como alternativa mais resistente a variações e tentativas de falsificação.

Para além da validação experimental, o trabalho permitiu identificar pontos críticos para melhoria, nomeadamente a necessidade de um pré-alinhamento geométrico e de técnicas mais robustas de filtragem de minúcias falsas, essenciais para otimizar o desempenho da componente morfológica. Estes desafios abrem caminho a desenvolvimentos futuros, como a integração de métodos automáticos de normalização de impressões digitais, a expansão do conjunto de dados e a utilização de técnicas de *machine learning* para ponderação adaptativa dos scores de fusão.

Em síntese, o sistema proposto demonstrou ser funcional, versátil e escalável, constituindo uma prova de conceito sólida para a utilização combinada de sinais fisiológicos e morfológicos em autenticação biométrica.

6.2 Trabalho Futuro

Futuramente, o sistema poderá ser expandido e otimizado em várias vertentes. Ao nível de processamento, recomenda-se a implementação de técnicas automáticas de pré-alinhamento geométrico e segmentação adaptativa, capazes de melhorar a correspondência entre amostras e reduzir o impacto de falsas minúcias.

Devido a limitações de tempo, não foi possível desenvolver um dispositivo compacto dedicado à autenticação bimodal, sendo necessária a utilização do CardioWheel [6], que apesar de ser uma inovação biométrica na condução, não constitui uma solução prática ou cómoda em contextos de segurança. Acresce ainda a utilização de um sensor ótico, que apresenta dimensões superiores às de um sensor capacitivo e requer uma maior dependência energética, o que limita a sua integração em dispositivos de menor volume.

Por fim recomenda-se a realização de testes específicos de segurança, avaliando a resistência do sistema a ataques de *spoofing* tanto em cada modalidade individual como no sistema multimodal. A concretização destes caminhos de investigação poderá viabilizar a aplicação do sistema em cenários reais de autenticação, como dispositivos móveis, controlo de acessos ou contextos médicos ou setores financeiros, entre outros, contribuindo para o desenvolvimento de soluções biométricas mais seguras, fiáveis e versáteis.

Referências Bibliográficas

[1] A. Al Abdulwahid, N. Clarke, I. Stengel, S. Furnell, e C. Reich, «Continuous and transparent multimodal authentication: reviewing the state of the art», *Cluster Comput*, vol. 19, n. 1, pp. 455–474, Mar. 2016, doi: 10.1007/s10586-015-0510-4.

[2] M. Be, A. M. Abhishek, T. J, V. K. R, and L. M. Patnaik, “Multimodal Biometric Authentication using ECG and Fingerprint,” International Journal of Computer Applications, vol. 111, no. 13, pp. 33–39, Feb. 2015, doi: 10.5120/19601-1452.

[3] N. Ammour, Y. Bazi, e N. Alajlan, «Multimodal Approach for Enhancing Biometric Authentication», *J Imaging*, vol. 9, n. 9, Set. 2023, doi: 10.3390/jimaging9090168.

[4] M. Komeili, N. Armanfard, e D. Hatzinakos, «Liveness Detection and Automatic Template Updating Using Fusion of ECG and Fingerprint», *IEEE Transactions on Information Forensics and Security*, vol. 13, n. 7, pp. 1810–1822, Jul. 2018, doi: 10.1109/TIFS.2018.2804890.

[5] J. S. Arteaga-Falconi, H. Al Osman, e A. El Saddik, «ECG and fingerprint bimodal authentication», *Sustain Cities Soc*, vol. 40, pp. 274–283, Jul. 2018, doi: 10.1016/j.scs.2017.12.023.

[6] A. Lourenço, A. Alves, C. Carreiras, R. Duarte, e A. Fred, *CardioWheel: ECG Biometrics on the Steering Wheel*, vol. 9286. em Lecture Notes in Computer Science, vol. 9286. Cham: Springer International Publishing, 2015. doi: 10.1007/978-3-319-23461-8.

[7] A. K. Jain, A. Ross, e S. Pankanti, «Biometrics: A tool for information security», *IEEE Transactions on Information Forensics and Security*, vol. 1, n. 2, pp. 125–143, Jun. 2006, doi: 10.1109/TIFS.2006.873653.

[8] A. Ross and A. K. Jain, “Multimodal biometrics: An overview,” European Signal Processing Conference, pp. 1221–1224, Sep. 2004, doi: 10.5281/zenodo.38715.

[9] A. K. Jain and A. Kumar, "Biometrics of Next Generation: An Overview," *Second Generation Biometrics*, Springer, 2010.

[10] A. K. Jain, P. Flynn, and A. A. Ross, *Handbook of Biometrics*. Springer, 2007. ISBN 978-0-387-71040-2.

[11] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, Handbook of Fingerprint Recognition. 2003. doi: 10.1007/b97303.

[12] J. D. Glover *et al.*, «The developmental basis of fingerprint pattern formation and variation», *Cell*, vol. 186, n. 5, pp. 940-956.e20, Mar. 2023, doi: 10.1016/j.cell.2023.01.015.

[13] S. S.Omran e M. Abdulmunem Salih, «Design and Implementation of Multi-model Biometric Identification System», *Int J Comput Appl*, vol. 99, n. 15, pp. 14–21, Ago. 2014, doi: 10.5120/17448-8255.

[14] Y. Yu, Q. Niu, X. Li, J. Xue, W. Liu, and D. Lin, “A review of Fingerprint sensors: Mechanism, Characteristics, and applications,” Micromachines, vol. 14, no. 6, p. 1253, Jun. 2023, doi: 10.3390/mi14061253.

[15] N. Martins, J. S. Silva, e A. Bernardino, «Fingerprint Recognition in Forensic Scenarios», *Sensors*, vol. 24, n. 2, Jan. 2024, doi: 10.3390/s24020664.

[16] A. J. Mohamed Abdul Cader, J. Banks, e V. Chandran, «Fingerprint Systems: Sensors, Image Acquisition, Interoperability and Challenges», *Sensors*, vol. 23, n. 14, Jul. 2023, doi: 10.3390/s23146591.

[17] R. German and K. S. Barber, "Current Biometric Adoption and Trends," Center for Identity, University of Texas at Austin, 2017. [Online] Available: https://identity.utexas.edu/sites/default/files/2020-09/Current%20Biometric%20Adoption%20and%20Trends.pdf

[18] Young-Hyun Baek, *ISOCC 2016 : International SoC Design Conference : «Smart SoC for Intelligent Things» : October 23-26, 2016, Ramada Plaza Jeju Hotel, Jeju, Korea*. IEEE, 2016.

[19] L. Qiu, "Fingerprint sensor technology," *2014 9th IEEE Conference on Industrial Electronics and Applications*, Hangzhou, China, 2014, pp. 1433-1436, doi: 10.1109/ICIEA.2014.6931393.

[20] D. Petrovska-Delacrétaz, G. Chollet, e B. Dorizzi, *Guide to biometric reference systems and performance evaluation*. Springer London, 2009. doi: 10.1007/978-1-84800-292-0.

[21] A. M. Bazen, G. T. B. Verwaaijen, S. H. Gerez, L. P. J. Veelenturf, and B. J. van der Zwaag, “A correlation-based fingerprint verification system,” in Proc. ProRISC 2000 Workshop on Circuits, Systems and Signal Processing, Veldhoven, The Netherlands, Nov. 2000.

[22] D. Peralta *et al.*, «A survey on fingerprint minutiae-based local matching for verification and identification: Taxonomy and experimental evaluation», *Inf Sci (N Y)*, vol. 315, pp. 67–87, Set. 2015, doi: 10.1016/j.ins.2015.04.013.

[23] A. N. Marana and A. K. Jain, "Ridge-Based Fingerprint Matching Using Hough Transform," *XVIII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'05)*, Natal, Brazil, 2005, pp. 112-119, doi: 10.1109/SIBGRAPI.2005.45.

[24] A. M. M. Chowdhury and M. H. Imtiaz, “Contactless Fingerprint Recognition Using Deep Learning—A Systematic Review,” Journal of Cybersecurity and Privacy, vol. 2, no. 3, pp. 714–730, Sep. 2022, doi: 10.3390/jcp2030036.

[25] M. Diarra, A. K. Jean, B. A. Bakary, e K. B. Medard, «Study of Deep Learning Methods f or Fingerprint Recognition», *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 10, n. 3, pp. 192–197, Set. 2021, doi: 10.35940/ijrte.C6478.0910321.

[26] I. Pinto, A. L. N. Fred, C. Rodrigues, and H. P. da Silva, Electrophysiology of the Heart and the Electrocardiogram: Visual Depictions, Tech. Rep., Instituto de Telecomunicações, Dec. 2020. [Online]. Available: https://www.researchgate.net/publication/346910888

[27] H. S. Choi, B. Lee, e S. Yoon, «Biometric Authentication Using Noisy Electrocardiograms Acquired by Mobile Sensors», *IEEE Access*, vol. 4, pp. 1266–1273, 2016, doi: 10.1109/ACCESS.2016.2548519.

[28] P. G. Malghan e M. K. Hota, «A review on ECG filtering techniques for rhythm analysis», 1 de Junho de 2020, *Springer*. doi: 10.1007/s42600-020-00057-9.

[29] M. Ingale, R. Cordeiro, S. Thentu, Y. Park, e N. Karimian, «ECG Biometric Authentication: A Comparative Analysis», *IEEE Access*, vol. 8, pp. 117853–117866, 2020, doi: 10.1109/ACCESS.2020.3004464.

[30] P. J. . Podrid, Rajeev. Malhotra, Rahul. Kakkar, e P. A. . Noseworthy, *Podrid’s real-world ECGs : a master’s approach to the art and practice of clinical ECG interpretation. Volume 1, The basics*. Cardiotext Publishing, 2013.

[31] A. Lourenço, H. Silva and A. Fred, "ECG-based biometrics: A real time classification approach," *2012 IEEE International Workshop on Machine Learning for Signal Processing*, Santander, Spain, 2012, pp. 1-6, doi: 10.1109/MLSP.2012.6349735.

[32] J. Ribeiro Pinto, J. S. Cardoso, e A. Lourenco, «Evolution, current challenges, and future possibilities in ECG Biometrics», 21 de Junho de 2018, *Institute of Electrical and Electronics Engineers Inc.* doi: 10.1109/ACCESS.2018.2849870.

[33] H. P. da Silva, A. Lourenço, A. Fred, and A. K. Jain, "Finger ECG signal for user authentication: usability and performance," IT - Instituto de Telecomunicações, IST -

Instituto Superior Técnico, ISEL - Instituto Superior de Engenharia de Lisboa, Portugal; Michigan State University, USA.

[34] H. S. and A. F. A. Lourenço, *ECG-based biometrics: A real time classification approach*. IEEE, 2012.

[35] J. Yoo, L. Yan, S. Lee, H. Kim, e H. J. Yoo, «A wearable ECG acquisition system with compact planar-fashionable circuit board-based shirt», *IEEE Transactions on Information Technology in Biomedicine*, vol. 13, n. 6, pp. 897–902, Nov. 2009, doi: 10.1109/TITB.2009.2033053.

[36] H. P. Da Silva, C. Carreiras, A. Lourenço, e A. Fred, «Off-the-person electrocardiography», em *Proceedings of the International Congress on Cardiovascular Technologies*, 2013, pp. 99–106. doi: 10.5220/0004647700990106.

[37] S. Nayak, M. K. Soni, and D. Bansal, "Filtering techniques for ECG signal processing," *Int. J. Res. Eng. Appl. Sci.*, vol. 2, no. 2, Feb. 2012. [Online]. Available: https://www.euroasiapub.org/wp-content/uploads/2012/02/10-1-1-470-1205.pdf

[38] Y. Kumar e G. K. Malik, «Performance Analysis of different filters for power line interface reduction in ECG signal», *Int J Comput Appl*, vol. 3, n. 7, pp. 1–6, Jun. 2010, doi: 10.5120/746-1055.

[39] D. P. Coutinho, H. Silva, H. Gamboa, A. Fred, e M. Figueiredo, «Novel fiducial and non-fiducial approaches to electrocardiogram-based biometric systems», *IET Biom*, vol. 2, n. 2, pp. 64–75, 2013, doi: 10.1049/iet-bmt.2012.0055.

[40] F. Canento, A. Lourenço, H. Silva, e A. Fred, «Review and Comparison of Real Time Electrocardiogram Segmentation Algorithms for Biometric Applications».

[41] S. Asadianfam, M. J. Talebi, e E. Nikougoftar, «ECG-based authentication systems: a comprehensive and systematic review», *Multimed Tools Appl*, vol. 83, n. 9, pp. 27647–27701, Mar. 2024, doi: 10.1007/s11042-023-16506-3.

[42] S. Chandra, A. Sharma, e G. K. Singh, «Feature extraction of ECG signal», *J Med Eng Technol*, vol. 42, n. 4, pp. 306–316, Mai. 2018, doi: 10.1080/03091902.2018.1492039.

[43] M. Hejazi, S. A. R. Al-Haddad, Y. P. Singh, S. J. Hashim, e A. F. Abdul Aziz, «ECG biometric authentication based on non-fiducial approach using kernel methods», *Digital Signal Processing: A Review Journal*, vol. 52, pp. 72–86, Mai. 2016, doi: 10.1016/j.dsp.2016.02.008.

[44] A. K. Singh and S. Krishnan, “ECG signal feature extraction trends in methods and applications,” BioMedical Engineering OnLine, vol. 22, no. 1, Mar. 2023, doi: 10.1186/s12938-023-01075-1.

[45] A. Lourenço, H. Silva, D. Perna Santos, and A. Fred, "Towards a finger based ECG biometric system," Instituto Superior de Engenharia de Lisboa, Instituto de Telecomunicações, Instituto Superior Técnico, Lisboa, Portugal.

[46] D. Meltzer and D. Luengo, “ECG-Based Biometric Recognition: A Survey of Methods and Databases,” Sensors, vol. 25, no. 6, p. 1864, Mar. 2025, doi: 10.3390/s25061864.

[47] N. Padkan, B. S. Bigham, and M. R. Faraji, “Fingerprint Matching using the Onion Peeling Approach and Turning Function,” arXiv (Cornell University), Jan. 2021, doi: 10.48550/arxiv.2110.00958.

[48] S. Bakheet, S. Alsubai, A. Alqahtani, e A. Binbusayyis, «Robust Fingerprint Minutiae Extraction and Matching Based on Improved SIFT Features», *Applied Sciences*, vol. 12, n. 12, Jun. 2022, doi: 10.3390/app12126122.

[49] V. Krivokuća and W. Abdulla, "Fast fingerprint alignment method based on minutiae orientation histograms," in Proc. 27th Conf. Image and Vision Computing New Zealand (IVCNZ '12), Nov. 2012, pp. 486–491. doi: 10.1145/2425836.2425928.

[50] A. M. Bahaa-Eldin, «A medium resolution fingerprint matching system», *Ain Shams Engineering Journal*, vol. 4, n. 3, pp. 393–408, 2013.

[51] Cuevas, “GitHub - cuevas1208/fingerprint_recognition: An implementations of fingerprint recognition algorithm,” GitHub, [Online]. Available: https://github.com/cuevas1208/fingerprint_recognition

[52] L. Biel, O. Pettersson, L. Philipson, e P. Wide, «ECG analysis: A new approach in human identification», *IEEE Trans Instrum Meas*, vol. 50, n. 3, pp. 808–812, Jun. 2001, doi: 10.1109/19.930458.

[53] H. Silva, H. Gamboa, and A. Fred, “One Lead ECG Based Personal Identification with Feature Subspace Ensembles,” in Lecture notes in computer science, 2007, pp. 770–783. doi: 10.1007/978-3-540-73499-4_58.

[54] D. P. Coutinho, A. L. N. Fred, e M. A. T. Figueiredo, «One-lead ECG-based personal identification using Ziv-Merhav cross parsing», em *Proceedings International Conference on Pattern Recognition*, 2010, pp. 3858–3861. doi: 10.1109/ICPR.2010.940.

[55] A. D. C. Chan, M. M. Hamdy, A. Badre, e V. Badee, «Wavelet distance measure for person identification using electrocardiograms», *IEEE Trans Instrum Meas*, vol. 57, n. 2, pp. 248–253, Fev. 2008, doi: 10.1109/TIM.2007.909996.

[56] S. Sarhan, S. Alhassan, e S. Elmougy, «Multimodal Biometric Systems: A Comparative Study», *Arab J Sci Eng*, vol. 42, n. 2, pp. 443–457, Fev. 2017, doi: 10.1007/s13369-016-2241-0.

[57] W. Ahmed, A. Dahea, W. Dahea, e H. S. Fadewar, «Multimodal biometric system: A review», *International Journal of Research in Advanced Engineering and Technology 25 International Journal of Research in Advanced Engineering and Technology*, vol. 4, pp. 2455–0876, 2018, doi: 10.13140/RG.2.2.34056.65287.

[58] W. Ahmed, A. Dahea, W. Dahea, and H. S. Fadewar, "Multimodal biometric system: A review," *Int. J. Res. Adv. Eng. Technol.*, vol. 4, 2018. doi: 10.13140/RG.2.2.34056.65287.

[59] M. Ghayoumi, "A review of multimodal biometric systems: Fusion methods and their applications," *2015 IEEE/ACIS 14th International Conference on Computer and Information Science (ICIS)*, Las Vegas, NV, USA, 2015, pp. 131-136, doi: 10.1109/ICIS.2015.7166582.

[60] Adafruit Industries, «Fingerprint Sensor», Acedido: 2 de Setembro de 2025. Disponível em: https://www.adafruit.com/product/751

[61] Adafruit Industries, «Adafruit_CircuitPython_Fingerprint», Acedido: 2 de Setembro de 2025. Disponível em: https://github.com/adafruit/Adafruit_CircuitPython_Fingerprint

[62] Raspberry Pi Foundation, «Raspberri Pi 5», Acedido: 2 de Setembro de 2025. [Em linha]. Disponível em: https://www.raspberrypi.com/products/raspberry-pi-5/

[63] «FTDI», Acedido: 2 de Setembro de 2025. Disponível em: https://mauser.pt/catalog/product_info.php?products_id=096-6949

[64] P. Bota, R. Silva, C. Carreiras, A. Fred, e H. P. da Silva, «BioSPPy: A Python toolbox for physiological signal processing», *SoftwareX*, vol. 26, Mai. 2024, doi: 10.1016/j.softx.2024.101712.

[65] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman and A. K. Jain, "FVC2002: Second Fingerprint Verification Competition," 2002 International Conference on Pattern Recognition, Quebec City, QC, Canada, 2002, pp. 811-814 vol.3, doi: 10.1109/ICPR.2002.1048144.

[66] Goldberger, A., Amaral, L., Glass, L., Hausdorff, J., Ivanov, P. C., Mark, R., ... & Stanley, H. E. (2000). PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. Circulation [Online]. 101 (23), pp. e215–e220. RRID:SCR_007345.

[67] K. R. Moses, P. Higgins, M. McCabe, S. Probhakar, and S. Swann, "Automated Fingerprint Identification System (AFIS)," in *Fingerprint Sourcebook*, NCJ Number 225326, Feb. 2011, pp. 1–33.

Anexos

#ecg_aquisition

```
import os
import datetime
from functools import partial
import asyncio
import signal
from bleak import BleakClient

ECG_CHAR = 'f334ccc4-55fe-01bf-e611-b9e4166680f9'
ADDRESS = "30:C9:22:9A:1C:1A"

def write_ecg_config(fid, ts, fs):
    fid.write(f"# Date:= {ts.isoformat()}\n")
    fid.write(f"# Sampling Rate (Hz):= {fs:0.2f}\n")
    fid.write("# Labels:= LOD\tECG\n")
    fid.flush()

def decode_write_ecg_signal(fid, data):
    for i in range(0, len(data), 2):
        if i + 1 >= len(data):
            break
        value = data[i] | (data[i+1] << 8)
        v_ecg = (value >> 4) & 0x0fff
        v_lod = value & 0x0f
        fid.write(f"{v_lod}\t{v_ecg}\n")
    fid.flush()

def handle_disconnect(client):
    print(f"Disconnected from device {client.address}.")

def handle_data(fid, handler, sender, data):
    try:
        handler(fid, data)
    except ValueError:
        pass

def handle_sigint(stop_evt, sig, frame=None):
    stop_evt.set()
    print("Stopping...")

async def acquire_ecg(address):
    path = os.path.join(os.getcwd(), "data")
    os.makedirs(path, exist_ok=True)

    stop_evt = asyncio.Event()
    signal.signal(signal.SIGINT, partial(handle_sigint, stop_evt))

    async with BleakClient(address, disconnected_callback=handle_disconnect) as
client:
        now = datetime.datetime.utcnow()
        now_str = now.strftime('%Y_%m_%dT%H_%M_%S')
        ecg_path = os.path.join(path, f"ecg_{now_str}.txt")

        with open(ecg_path, 'w') as ecg_f:
            write_ecg_config(ecg_f, now, 1000)
            await  client.start_notify(ECG_CHAR,  partial(handle_data,  ecg_f,
decode_write_ecg_signal))
            print("A gravar ECG... Ctrl+C para terminar.")
            await stop_evt.wait()
```

```
    print(f"ECG guardado em: {ecg_path}")
    return ecg_path

def acquire_ecg_and_save():
    async def run_acquisition():
        return await acquire_ecg(ADDRESS)
    return asyncio.run(run_acquisition())
```

#ecg_processing

```
import os
import json
import numpy as np
from biosppy.signals import ecg
from scipy.signal import resample
from scipy.spatial.distance import cosine

RAW_DIR = "data"
PROCESSED_DIR = os.path.join(RAW_DIR, "processed")

def normalize_templates(templates, target_length=200):
    normalized = []
    for t in templates:
        t_resampled = resample(t, target_length)
        t_norm = (t_resampled - np.mean(t_resampled)) / (np.std(t_resampled) +
1e-6)
        normalized.append(t_norm.tolist())
    return normalized

def compute_mean_template(templates):
    return np.mean(np.array(templates), axis=0).tolist()

def process_txt_ecg(file_path, sampling_rate=1000):
    try:
        data = np.loadtxt(file_path, comments="#", usecols=1)
        if len(data) < 5000:
            print("? ECG demasiado curto. Mnimo recomendado: 5000 amostras.")
            return None

        output = ecg.ecg(signal=data, sampling_rate=sampling_rate, show=False)

        print(f"? ECG carregado: {file_path}")
        print("   N de amostras:", len(data))
        print("   N de templates extrados:", len(output["templates"]))

        templates_raw = output["templates"]
        templates_norm = normalize_templates(templates_raw)
        mean_template = compute_mean_template(templates_norm)

        return {
            "templates": templates_norm,
            "mean_template": mean_template
        }

    except Exception as e:
        print(f"? Erro ao processar ECG: {e}")
        return None


def save_processed_template(user_id, template, tipo):
    folder = f"Person_{user_id:02d}"
    out_dir = os.path.join(PROCESSED_DIR, folder)
    os.makedirs(out_dir, exist_ok=True)
    out_path = os.path.join(out_dir, f"{tipo}_template.json")
    with open(out_path, "w", encoding="utf-8") as f:
        json.dump(template, f, indent=4)
    print(f"? Template guardado em {out_path}")
```

```
def authenticate_user(template, user_id, threshold=0.80):
    folder = f"Person_{user_id:02d}"
    enrolled_path         =         os.path.join(PROCESSED_DIR,         folder,
"enrolled_template.json")

    if not os.path.exists(enrolled_path):
        print("? Template enrolled no encontrado para este utilizador.")
        return

    try:
        with open(enrolled_path, "r") as f:
            enrolled = json.load(f)["mean_template"]
        sim          =          1          -          cosine(np.array(enrolled),
np.array(template["mean_template"]))
        print(f"Similaridade cosseno: {sim:.4f}")
        if sim >= threshold:
            print("? Autenticacao bem-sucedida!")
        else:
            print("? Autenticacao falhou.")
    except Exception as e:
        print(f"? Erro na autenticao: {e}")



def identify_user(template, threshold=0.80):
    try:
        test = template["mean_template"]
        max_sim = -1
        identified_user = None
        for folder in os.listdir(PROCESSED_DIR):
            path          =          os.path.join(PROCESSED_DIR,          folder,
"enrolled_template.json")
            if not os.path.exists(path):
                continue
            with open(path, "r") as f2:
                enrolled = json.load(f2)["mean_template"]
            sim = 1 - cosine(enrolled, test)
            if sim > max_sim:
                max_sim = sim
                identified_user = folder
        if max_sim >= threshold:
            print(f"?   Identificado   como   {identified_user}   (similaridade:
{max_sim:.4f})")
        else:
            print(f"? Nenhuma correspondncia encontrada. Similaridade mxima:
{max_sim:.4f}")
    except Exception as e:
        print(f"Erro na identificao: {e}")
```

#ecg_processing2

```
import os
import json
import wfdb
import numpy as np
from biosppy.signals import ecg as ecg_bio
from scipy.signal import resample
from scipy.spatial.distance import cosine

RAW_DIR = "data"
PROCESSED_DIR = os.path.join(RAW_DIR, "processed")

def _normalize_templates(templates, target_length=200):
    out = []
    for t in templates:
```

```
        tr = resample(t, target_length)
        tr = (tr - np.mean(tr)) / (np.std(tr) + 1e-6)
        out.append(tr.tolist())
    return out

def compute_mean_template(templates):
    return np.mean(np.array(templates), axis=0).tolist()

def process_wfdb_ecg(rec_base_path, lead=0):
    rec = wfdb.rdrecord(rec_base_path)
    fs = rec.fs
    sig = rec.p_signal[:, lead].astype(float)
    out = ecg_bio.ecg(signal=sig, sampling_rate=fs, show=False)
    templates = out["templates"]
    if len(templates) == 0:
        print("[!] nenhum template ECG encontrado")
        return None
    templ_norm = _normalize_templates(templates)
    mean_t = compute_mean_template(templ_norm)
    return {"templates": templ_norm, "mean_template": mean_t}

def save_processed_template(user_id, template, tipo):
    folder = f"Person_{user_id:02d}"
    out_dir = os.path.join(PROCESSED_DIR, folder)
    os.makedirs(out_dir, exist_ok=True)
    out_path = os.path.join(out_dir, f"{tipo}_template.json")
    with open(out_path, "w", encoding="utf-8") as f:
        json.dump(template, f, indent=4)
    print(f"[ok] template guardado em {out_path}")

def authenticate_user(template, user_id, threshold=0.80):
    folder = f"Person_{user_id:02d}"
    enrolled_path         =         os.path.join(PROCESSED_DIR,         folder,
"enrolled_template.json")
    if not os.path.exists(enrolled_path):
        print("[!] template enrolled nao encontrado para este utilizador")
        return
    try:
        with open(enrolled_path, "r") as f:
            enrolled = json.load(f)["mean_template"]
        sim         =         1         -         cosine(np.array(enrolled),
np.array(template["mean_template"]))
        print(f"similaridade cosseno: {sim:.4f}")
        if sim >= threshold:
            print("[?] autenticacao bem-sucedida")
        else:
            print("[?] autenticacao falhou")
    except Exception as e:
        print(f"[!] erro na autenticacao: {e}")

def identify_user(template, threshold=0.80):
    try:
        test = template["mean_template"]
        max_sim = -1.0
        identified_user = None
        for folder in os.listdir(PROCESSED_DIR):
            path         =         os.path.join(PROCESSED_DIR,         folder,
"enrolled_template.json")
            if not os.path.exists(path):
                continue
            with open(path, "r") as f2:
                enrolled = json.load(f2)["mean_template"]
            sim = 1 - cosine(enrolled, test)
            if sim > max_sim:
                max_sim = sim
                identified_user = folder
        if max_sim >= threshold:
```

```
            print(f"[?]    identificado    como    {identified_user}    (sim:
{max_sim:.4f})")
        else:
            print(f"[?] nenhuma correspondencia. sim max: {max_sim:.4f}")
    except Exception as e:
        print(f"[!] erro na identificacao: {e}")
```

 # menumultimodal.py

```
import os
import sys
import json
import time
import csv
from glob import glob
import numpy as np
import cv2 as cv
import matplotlib.pyplot as plt
from PIL import Image
import serial
import adafruit_fingerprint
from scipy.spatial.distance import cosine
from codigoECG.ecg_acquisition import acquire_ecg_and_save
from codigoECG.ecg_processing import process_txt_ecg
from utils.crossing_number import calculate_minutiaes
from utils.normalization import normalize
from utils.segmentation import create_segmented_and_variance_images
from utils.gabor_filter import gabor_filter
from utils.frequency import ridge_freq
from utils import orientation
from utils.skeletonize import skeletonize

RESULTADOS_DIR  = "resultados"
COMBINED_DIR    = "combined_templates"
os.makedirs(RESULTADOS_DIR, exist_ok=True)
os.makedirs(COMBINED_DIR,   exist_ok=True)

MAX_TRACKS   = 14
TRACK_WIDTH  = 10
N_SAMPLES_PER_USER  = 3
TEMPLATE_SAMPLE_IDX = 1
W_ECG = 0.7
W_FP  = 0.3
FP_T1 = 27
FP_T2 = 18
UART_DEV = "/dev/ttyUSB0"
uart   = serial.Serial(UART_DEV, baudrate=57600, timeout=1)
finger = adafruit_fingerprint.Adafruit_Fingerprint(uart)

def estimate_core_point(minutiae):
    if not minutiae:
        raise ValueError("lista de minucias vazia")
    xs = [m["x"] for m in minutiae]
    ys = [m["y"] for m in minutiae]
    return int(np.mean(xs)), int(np.mean(ys))

def preprocess_image_for_minutiae(img_gray):
    block_size = 16
    norm_img = normalize(img_gray, float(100), float(100))
    _,    normim,    mask    =    create_segmented_and_variance_images(norm_img,
block_size, 0.2)
    angles = orientation.calculate_angles(norm_img, W=block_size, smoth=False)
    freq   =   ridge_freq(normim,   mask,   angles,   block_size,   kernel_size=5,
minWaveLength=5, maxWaveLength=15)
    gabor_img = gabor_filter(normim, angles, freq)
    thin_image = skeletonize(gabor_img)
    return thin_image
```

```
def save_fingerprint_image(filename):
    while finger.get_image():
        pass
    img = Image.new("L", (256, 288), "white")
    pixeldata = img.load()
    mask = 0b00001111
    result = finger.get_fpdata(sensorbuffer="image")
    x = 0
    y = 0
    for i in range(len(result)):
        pixeldata[x, y] = (int(result[i]) >> 4) * 17
        x += 1
        pixeldata[x, y] = (int(result[i]) & mask) * 17
        if x == 255:
            x = 0
            y += 1
        else:
            x += 1
    img.save(filename)

def       compute_feature_vector(minutiae,       core_x=None,       core_y=None,
track_width=TRACK_WIDTH, max_tracks=MAX_TRACKS):
    if not minutiae:
        raise ValueError("minucias vazias")
    if core_x is None or core_y is None:
        core_x, core_y = estimate_core_point(minutiae)
    vector = np.zeros((max_tracks, 2), dtype=int)
    for m in minutiae:
        x, y, tipo = m["x"], m["y"], m["tipo"]
        r = int(np.hypot(x - core_x, y - core_y) // track_width)
        if r < max_tracks:
            vector[r][0 if tipo == 1 else 1] += 1
    return vector

def compare_fingerprints(v1, v2):
    v1 = np.array(v1)
    v2 = np.array(v2)
    length = min(len(v1), len(v2))
    v1, v2 = v1[:length], v2[:length]
    diff = np.abs(v1 - v2)
    sum1 = np.sum(diff[:, 0])
    sum2 = np.sum(diff[:, 1])
    return sum1, sum2

def fp_similarity_normalized(sum1, sum2, t1=FP_T1, t2=FP_T2):
    d = (sum1 / t1) + (sum2 / t2)
    return 1.0 / (1.0 + d)

def fuse_scores(score_ecg, score_fp):
    return 0.7*score_ecg + 0.3*score_fp

def save_combined_template(user_id, ecg_template, vec_fp, sample_id=None):
    if sample_id is None:
        raise ValueError("sample_id obrigatorio")
    fname = f"user_{user_id:02d}_s{sample_id}_combined.json"
    path = os.path.join(COMBINED_DIR, fname)
    data = {
        "ecg": ecg_template["mean_template"],
        "fp":  np.asarray(vec_fp).tolist()
    }
    with open(path, "w") as f:
        json.dump(data, f, indent=2)
    print(f"[ok] template guardado: {path}")

def load_combined_templates_s1():
    templates = []
    for file in sorted(os.listdir(COMBINED_DIR)):
```

```
        if not file.endswith("_s1_combined.json"):
            continue
        path = os.path.join(COMBINED_DIR, file)
        with open(path, "r") as f:
            data = json.load(f)
        templates.append((file, np.array(data["ecg"], dtype=float),
                          np.array(data["fp"],  dtype=float)))
    return templates

def register_multimodal():
    print(">> registo multimodal: 3 amostras; s1 sera o template oficial")
    try:
        user_id = int(input("ID do utilizador: ").strip())
    except ValueError:
        print("[!] ID invalido"); return
    for s in range(1, N_SAMPLES_PER_USER + 1):
        print(f"\n>> amostra {s}/3 - adquirir ECG...")
        ecg_path = acquire_ecg_and_save()
        ecg_template = process_txt_ecg(ecg_path)
        if not ecg_template:
            print("[!] falha no processamento do ECG"); return
        print(">> colocar o dedo no sensor (fingerprint)...")
        img_path = os.path.join(RESULTADOS_DIR, f"user_{user_id:02d}_s{s}.png")
        save_fingerprint_image(img_path)
        img_gray = cv.imread(img_path, cv.IMREAD_GRAYSCALE)
        thin = preprocess_image_for_minutiae(img_gray)
        mins = calculate_minutiaes(thin)
        core_x, core_y = estimate_core_point(mins)
        vec_fp = compute_feature_vector(mins, core_x, core_y)
        save_combined_template(user_id, ecg_template, vec_fp, sample_id=s)
        anotada = np.full_like(thin, 255); anotada[thin == 0] = 0
        anotada = cv.cvtColor(anotada, cv.COLOR_GRAY2BGR)
        for r in range(1, MAX_TRACKS):
            cv.circle(anotada, (core_x, core_y), r * TRACK_WIDTH, (200, 200,
200), 1)
        for m in mins:
            cor = (0, 0, 255) if m["tipo"] == 1 else (0, 255, 0)
            cv.circle(anotada, (int(m["x"]), int(m["y"])), 2, cor, 1)
        out_min                  =                 os.path.join(COMBINED_DIR,
f"minucias_output_{user_id:02d}_s{s}.png")
        cv.imwrite(out_min, anotada)
        plot_dir       =       os.path.join(COMBINED_DIR,       "ecg_plots");
os.makedirs(plot_dir, exist_ok=True)
        plot_path                    =                   os.path.join(plot_dir,
f"user_{user_id:02d}_s{s}_ecg_mean.png")
        plt.figure(figsize=(6, 2)); plt.plot(ecg_template["mean_template"])
        plt.title(f"Template  medio  ECG  (s{s})");  plt.xlabel("amostras");
plt.ylabel("amp norm")
        plt.tight_layout(); plt.savefig(plot_path); plt.close()
        if s < N_SAMPLES_PER_USER:
            print(">> aguarde 3-5 segundos..."); time.sleep(5)
    print(f"\n[ok] registo concluido para user {user_id:02d} (s1,s2,s3)")

def identify_multimodal():
    ecg_path = acquire_ecg_and_save()
    ecg_template = process_txt_ecg(ecg_path)
    if not ecg_template:
        print("[!] falha no processamento do ECG"); return
    tmp_path = os.path.join(RESULTADOS_DIR, "temp_test.png")
    save_fingerprint_image(tmp_path)
    img_gray = cv.imread(tmp_path, cv.IMREAD_GRAYSCALE)
    thin = preprocess_image_for_minutiae(img_gray)
    min_test = calculate_minutiaes(thin)
    cx_t, cy_t = estimate_core_point(min_test)
    vec_fp_test = compute_feature_vector(min_test, cx_t, cy_t)
    templates = load_combined_templates_s1()
    if not templates:
        print("[!] nenhum template s1 encontrado"); return
```

```
    best_name, best_score = None, -1.0
    for name, ecg_vec, fp_vec in templates:
        score_ecg = 1.0 - cosine(ecg_template["mean_template"], ecg_vec)
        s1, s2 = compare_fingerprints(vec_fp_test, fp_vec)
        score_fp = fp_similarity_normalized(s1, s2, FP_T1, FP_T2)
        score_fusao = fuse_scores(score_ecg, score_fp)
        print(f"{name}:    ECG={score_ecg:.3f}    |    FP={score_fp:.3f}    |
Fusao={score_fusao:.3f}")
        if score_fusao > best_score:
            best_score, best_name = score_fusao, name
    TH = 0.71
    if best_score >= TH:
        print(f"[?] Identificado como {best_name} com score {best_score:.4f}")
    else:
        print("[?] Nenhuma correspondencia acima do limiar")

def  registar_tentativa(user_id_input,  user_id_real,  score_ecg,  score_fp,
score_fusao, sucesso):
    csv_path = os.path.join(RESULTADOS_DIR, "tentativas.csv")
    file_exists = os.path.exists(csv_path)
    with open(csv_path, "a", newline="") as f:
        writer = csv.writer(f)
        if not file_exists:
            writer.writerow(["user_input",      "user_real",      "score_ecg",
"score_fp", "score_fusao", "sucesso", "timestamp"])
        writer.writerow([
            user_id_input,
            user_id_real,
            f"{score_ecg:.4f}",
            f"{score_fp:.4f}",
            f"{score_fusao:.4f}",
            "sim" if sucesso else "nao",
            time.strftime("%Y-%m-%d %H:%M:%S")
        ])

def authenticate_multimodal():
    try:
        user_id = int(input("ID do utilizador a autenticar: ").strip())
    except ValueError:
        print("[!] ID invalido"); return
    tpl_path                    =                 os.path.join(COMBINED_DIR,
f"user_{user_id:02d}_s1_combined.json")
    if not os.path.exists(tpl_path):
        print("[!] template s1 nao encontrado para este user"); return
    ecg_path = acquire_ecg_and_save()
    ecg_template = process_txt_ecg(ecg_path)
    if not ecg_template:
        print("[!] falha no processamento do ECG"); return
    tmp_path = os.path.join(RESULTADOS_DIR, "temp_test.png")
    save_fingerprint_image(tmp_path)
    img_gray = cv.imread(tmp_path, cv.IMREAD_GRAYSCALE)
    thin = preprocess_image_for_minutiae(img_gray)
    min_test = calculate_minutiaes(thin)
    cx_t, cy_t   = estimate_core_point(min_test)
    vec_fp_test  = compute_feature_vector(min_test, cx_t, cy_t)
    with open(tpl_path, "r") as f:
        data = json.load(f)
    enrolled_ecg = np.array(data["ecg"], dtype=float)
    enrolled_fp  = np.array(data["fp"],  dtype=float)
    score_ecg = 1.0 - cosine(ecg_template["mean_template"], enrolled_ecg)
    s1, s2    = compare_fingerprints(vec_fp_test, enrolled_fp)
    score_fp  = fp_similarity_normalized(s1, s2, FP_T1, FP_T2)
    score_fus = fuse_scores(score_ecg, score_fp)
    print(f"User  {user_id:02d}  |  ECG={score_ecg:.4f}  |  FP={score_fp:.4f}  |
Fusao={score_fus:.4f}")
    TH = 0.71
    if score_fus >= TH:
        print("[?] Autenticado com sucesso.")
```

```
        registar_tentativa(user_id, user_id, score_ecg, score_fp, score_fus,
True)
    else:
        print("[?] Autenticacao falhou.")
        registar_tentativa(user_id, user_id, score_ecg, score_fp, score_fus,
False)

def limpar_templates():
    confirm = input("Tens a certeza que queres apagar TODOS os registos? (s/n):
").strip().lower()
    if confirm != "s":
        print("Operacao cancelada."); return
    if finger.empty_library() == adafruit_fingerprint.OK:
        print("[ok] templates do sensor apagados")
    else:
        print("[!] erro ao apagar templates do sensor")
    for root, _, files in os.walk(RESULTADOS_DIR):
        for f in files:
            if    f.startswith(("minucias_",    "user_",    "temp_test"))    or
f.endswith(".json") or f.endswith(".png"):
                try:
                    os.remove(os.path.join(root, f))
                except Exception:
                    pass
    for root, _, files in os.walk(COMBINED_DIR):
        for f in files:
            if f.endswith(".json") or f.endswith(".png"):
                try:
                    os.remove(os.path.join(root, f))
                except Exception:
                    pass
    print("[ok] ficheiros locais removidos")

def menu():
    while True:
        print("\n===== SISTEMA MULTIMODAL (ECG + Impressao Digital) =====")
        print("1 - Registo")
        print("2 - Identificacao (1:N)")
        print("3 - Autenticacao (1:1)")
        print("4 - Limpar todos os registos")
        print("0 - Sair")
        op = input("Escolha uma opcao: ").strip()
        if op == "0":
            break
        elif op == "1":
            register_multimodal()
        elif op == "2":
            identify_multimodal()
        elif op == "3":
            authenticate_multimodal()
        elif op == "4":
            limpar_templates()
        else:
            print("Opcao invalida.")

if __name__ == "__main__":
    menu()
```

 # build_from_dataset.py

```
import os
import json
import re
import numpy as np
import cv2 as cv
from ecg_processing2 import process_wfdb_ecg
from utils.crossing_number import calculate_minutiaes
```

```
from utils.normalization import normalize
from utils.segmentation import create_segmented_and_variance_images
from utils.gabor_filter import gabor_filter
from utils.frequency import ridge_freq
from utils import orientation
from utils.skeletonize import skeletonize

ECG_DIR                                                                    =
"/home/joaorcustodio/fingerprint_recognition/fusion/ecgdatabase/ecgdatabasetes
te"
FP_DIR  = "/home/joaorcustodio/fingerprint_recognition/fusion/ID"
COMBINED_DIR = "combined_templates"
os.makedirs(COMBINED_DIR, exist_ok=True)
MAX_TRACKS  = 5
TRACK_WIDTH = 31

def preprocess_image_for_minutiae(img_gray):
    block_size = 16
    norm_img = normalize(img_gray, float(100), float(100))
    _,    normim,    mask    =    create_segmented_and_variance_images(norm_img,
block_size, 0.2)
    angles = orientation.calculate_angles(norm_img, W=block_size, smoth=False)
    freq   =   ridge_freq(normim,   mask,   angles,   block_size,   kernel_size=5,
minWaveLength=5, maxWaveLength=15)
    gabor_img = gabor_filter(normim, angles, freq)
    return skeletonize(gabor_img)

def estimate_core_point(minutiae):
    xs = [m["x"] for m in minutiae]
    ys = [m["y"] for m in minutiae]
    return int(np.mean(xs)), int(np.mean(ys))

def  compute_feature_vector(minutiae,  core_x,  core_y,  track_width=TRACK_WIDTH,
max_tracks=MAX_TRACKS):
    vector = np.zeros((max_tracks, 2), dtype=int)
    for m in minutiae:
        x, y, tipo = m["x"], m["y"], m["tipo"]
        r = int(np.hypot(x - core_x, y - core_y) // track_width)
        if r < max_tracks:
            vector[r][0 if tipo == 1 else 1] += 1
    return vector

def save_combined(user_id, s_idx, ecg_template, fp_vec):
    out = {
        "ecg": np.asarray(ecg_template["mean_template"]).tolist(),
        "fp":  np.asarray(fp_vec).tolist()
    }
    fname = f"user_{int(user_id):02d}_s{s_idx}_combined.json"
    with open(os.path.join(COMBINED_DIR, fname), "w") as f:
        json.dump(out, f, indent=2)
    print(f"[ok] {fname} gerado")

CUSTOM_MAP = {1: 101, 2: 102, 3: 103, 4: 104, 5: 105, 6: 106, 7: 107, 8: 108,
9: 109, 10: 110}

def guess_fp_id(user_id, fp_index):
    if user_id in CUSTOM_MAP:
        return CUSTOM_MAP[user_id]
    cand = 100 + user_id
    if cand in fp_index:
        return cand
    if user_id in fp_index:
        return user_id
    return None

def build_fp_index():
    idx = {}
    for f in os.listdir(FP_DIR):
```

```
        m = re.match(r"^(\d{1,3})_([123])\.tif$", f)
        if not m:
            continue
        fid = int(m.group(1))
        s   = int(m.group(2))
        idx.setdefault(fid, set()).add(s)
    return idx

def dataset_ecg_path(user_id, s_idx):
    folder = os.path.join(ECG_DIR, f"Person_{int(user_id):02d}")
    return os.path.join(folder, f"rec_{s_idx}")

def dataset_fp_path(fp_id, s_idx):
    cands = [
        os.path.join(FP_DIR, f"{int(fp_id)}_{s_idx}.tif"),
        os.path.join(FP_DIR, f"{int(fp_id):02d}_{s_idx}.tif"),
        os.path.join(FP_DIR, f"{int(fp_id):03d}_{s_idx}.tif"),
    ]
    for c in cands:
        if os.path.exists(c):
            return c
    return None

def build_all(sessions=(1,2,3)):
    persons = sorted([d for d in os.listdir(ECG_DIR) if d.startswith("Person_")
and os.path.isdir(os.path.join(ECG_DIR, d))])
    if not persons:
        print("[erro] nao encontrei pastas Person_xx em", ECG_DIR)
        return
    fp_index = build_fp_index()
    if not fp_index:
        print("[erro] nao encontrei .tif em", FP_DIR)
        return
    total = 0
    for p in persons:
        user_id = int(p.split("_")[1])
        fp_id = guess_fp_id(user_id, fp_index)
        if fp_id is None:
            print(f"[skip] sem mapping FP para Person_{user_id:02d}")
            continue
        for s in sessions:
            rec_base = dataset_ecg_path(user_id, s)
            if     not     (os.path.exists(rec_base     +     ".hea")     and
os.path.exists(rec_base + ".dat")):
                print(f"[skip] ECG ausente: {user_id}_{s}")
                continue
            ecg_t = process_wfdb_ecg(rec_base)
            if not ecg_t:
                print(f"[skip] falha ECG: {user_id}_{s}")
                continue
            fp_path = dataset_fp_path(fp_id, s)
            if not fp_path:
                print(f"[skip] FP ausente: {fp_id}_{s}")
                continue
            img_gray = cv.imread(fp_path, cv.IMREAD_GRAYSCALE)
            thin = preprocess_image_for_minutiae(img_gray)
            mins = calculate_minutiaes(thin)
            if not mins:
                print(f"[skip] sem minucias: {fp_id}_{s}")
                continue
            cx, cy = estimate_core_point(mins)
            vec_fp = compute_feature_vector(mins, cx, cy)
            save_combined(user_id, s, ecg_t, vec_fp)
            total += 1
    print(f"[done] {total} templates combinados gerados")

if __name__ == "__main__":
    build_all()
```

 # evaluate_system.py

```
import os
import json
import csv
import numpy as np
from scipy.spatial.distance import cosine
import argparse
import matplotlib
matplotlib.use("Agg")
import matplotlib.pyplot as plt

COMBINED_DIR    = "combined_templates"
COMPARISONS_CSV = "scores_all_pairs.csv"
ROTATING_ENROLLMENT = True
ENROLL_SESSION      = "s1"
W_ECG = 0.7
W_FP  = 0.3
FP_T1 = 27.0
FP_T2 = 18.0

parser = argparse.ArgumentParser(description="Recebe parametros FP_T1 e FP_T2")
parser.add_argument("--FP_T1", type=float, help="Parametro FP_T1")
parser.add_argument("--FP_T2", type=float, help="Parametro FP_T2")
args = parser.parse_args()
if args.FP_T1:
    FP_T1 = args.FP_T1
if args.FP_T2:
    FP_T2 = args.FP_T2
print("FP_T1 =", FP_T1)
print("FP_T2 =", FP_T2)

def compare_fingerprints(v1, v2):
    v1 = np.array(v1)
    v2 = np.array(v2)
    L = min(len(v1), len(v2))
    v1, v2 = v1[:L], v2[:L]
    diff = np.abs(v1 - v2)
    sum1 = np.sum(diff[:, 0])
    sum2 = np.sum(diff[:, 1])
    return sum1, sum2

def fp_similarity_normalized(sum1, sum2, t1=FP_T1, t2=FP_T2):
    d = (sum1 / t1) + (sum2 / t2)
    return 1.0 / (1.0 + d)

def sim_ecg(a, b):
    return 1.0 - cosine(a, b)

def fuse_scores(se, sf):
    return 0.7*se + 0.3*sf

def _available_sessions(sess_dict):
    return [s for s in ["s1", "s2", "s3"] if s in sess_dict]

def _load_user_samples():
    samples = {}
    for f in sorted(os.listdir(COMBINED_DIR)):
        if not f.endswith("_combined.json") or "_s" not in f:
            continue
        parts = f.split("_")
        try:
            u = int(parts[1])
            s = parts[2]
        except Exception:
```

```
            continue
        path = os.path.join(COMBINED_DIR, f)
        with open(path, "r") as fh:
            d = json.load(fh)
        ecg = np.array(d["ecg"], dtype=float)
        fp  = np.array(d["fp"], dtype=float)
        samples.setdefault(u, {})[s] = (ecg, fp)
    return samples

def build_all_comparisons(csv_path=COMPARISONS_CSV, w=W_ECG):
    samples = _load_user_samples()
    users   = sorted(samples.keys())
    rows    = []
    if not users:
        print("[!] nenhum template encontrado")
        return
    for u in users:
        sess_u = _available_sessions(samples[u])
        if len(sess_u) < 2:
            continue
        if ROTATING_ENROLLMENT:
            pairs = [("s1","s2"), ("s2","s3"), ("s3","s1")]
            valid_pairs = [(t,p) for (t,p) in pairs if (t in sess_u and p in
sess_u)]
            for t_sess, p_sess in valid_pairs:
                ecg_t, fp_t = samples[u][t_sess]
                ecg_p, fp_p = samples[u][p_sess]
                se = sim_ecg(ecg_t, ecg_p)
                s1, s2 = compare_fingerprints(fp_t, fp_p)
                sf = fp_similarity_normalized(s1, s2, FP_T1, FP_T2)
                rows.append({
                    "template": u, "template_session": t_sess,
                    "probe": u, "probe_session": p_sess,
                    "tipo": "genuino",
                    "score_ecg":    se,    "score_fp":    sf,    "score_fusao":
fuse_scores(se, sf)
                    })
            for t_sess, _ in valid_pairs:
                ecg_t, fp_t = samples[u][t_sess]
                for v in users:
                    if v == u:
                        continue
                    for s_probe in _available_sessions(samples[v]):
                        ecg_p, fp_p = samples[v][s_probe]
                        se = sim_ecg(ecg_t, ecg_p)
                        s1, s2 = compare_fingerprints(fp_t, fp_p)
                        sf = fp_similarity_normalized(s1, s2, FP_T1, FP_T2)
                        rows.append({
                            "template": u, "template_session": t_sess,
                            "probe": v, "probe_session": s_probe,
                            "tipo": "impostor",
                            "score_ecg":  se,  "score_fp":  sf,  "score_fusao":
fuse_scores(se, sf)
                        })
        else:
            if ENROLL_SESSION not in sess_u:
                continue
            t_sess = ENROLL_SESSION
            ecg_t, fp_t = samples[u][t_sess]
            for p_sess in sess_u:
                if p_sess == t_sess:
                    continue
                ecg_p, fp_p = samples[u][p_sess]
                se = sim_ecg(ecg_t, ecg_p)
                s1, s2 = compare_fingerprints(fp_t, fp_p)
                sf = fp_similarity_normalized(s1, s2, FP_T1, FP_T2)
                rows.append({
                    "template": u, "template_session": t_sess,
```

```
                    "probe": u, "probe_session": p_sess,
                    "tipo": "genuino",
                    "score_ecg":    se,    "score_fp":    sf,    "score_fusao":
fuse_scores(se, sf)
                })
            for v in users:
                if v == u:
                    continue
                for s_probe in _available_sessions(samples[v]):
                    ecg_p, fp_p = samples[v][s_probe]
                    se = sim_ecg(ecg_t, ecg_p)
                    s1, s2 = compare_fingerprints(fp_t, fp_p)
                    sf = fp_similarity_normalized(s1, s2, FP_T1, FP_T2)
                    rows.append({
                        "template": u, "template_session": t_sess,
                        "probe": v, "probe_session": s_probe,
                        "tipo": "impostor",
                        "score_ecg":   se,   "score_fp":   sf,   "score_fusao":
fuse_scores(se, sf)
                    })
    if not rows:
        print("[!] sem comparacoes geradas")
        return
    with open(csv_path, "w", newline="") as f:
        wtr = csv.DictWriter(f, fieldnames=rows[0].keys())
        wtr.writeheader()
        wtr.writerows(rows)
    print(f"[ok] CSV criado: {csv_path} ({len(rows)} comparacoes)")
    g = sum(1 for r in rows if r["tipo"] == "genuino")
    i = sum(1 for r in rows if r["tipo"] == "impostor")
    print(f"     genuinos: {g} | impostores: {i}")

def _far_frr(gen, imp, ths):
    imp = np.array(imp)
    gen = np.array(gen)
    if len(imp) == 0 or len(gen) == 0:
        return 0.0, 0.0, 0.0, float("nan")
    fars = [np.sum(imp >= T) / len(imp) for T in ths]
    frrs = [np.sum(gen <  T) / len(gen) for T in ths]
    idx  = int(np.argmin(np.abs(np.array(fars) - np.array(frrs))))
    eer  = (fars[idx] + frrs[idx]) / 2.0
    return fars[idx], frrs[idx], eer, ths[idx]

def evaluate_csv(csv_path=COMPARISONS_CSV):
    with open(csv_path, "r") as f:
        rows = list(csv.DictReader(f))
    cols = ["score_ecg", "score_fp", "score_fusao"]
    for col in cols:
        gen = [float(x[col]) for x in rows if x["tipo"] == "genuino"]
        imp = [float(x[col]) for x in rows if x["tipo"] == "impostor"]
        ths = np.linspace(min(gen+imp), max(gen+imp), 400)
        FAR, FRR, EER, T = _far_frr(gen, imp, ths)
        print(f"{col:>12}: EER={EER*100:.2f}% @T={T:.4f} | FAR={FAR*100:.2f}% |
FRR={FRR*100:.2f}% | FP_T1={FP_T1} | FP_T2={FP_T2}")

if __name__ == "__main__":
    build_all_comparisons()
    evaluate_csv()
```