

Segurança da Informação

Competências:

- Implementar práticas e condutas de segurança da informação no ambiente de TI.

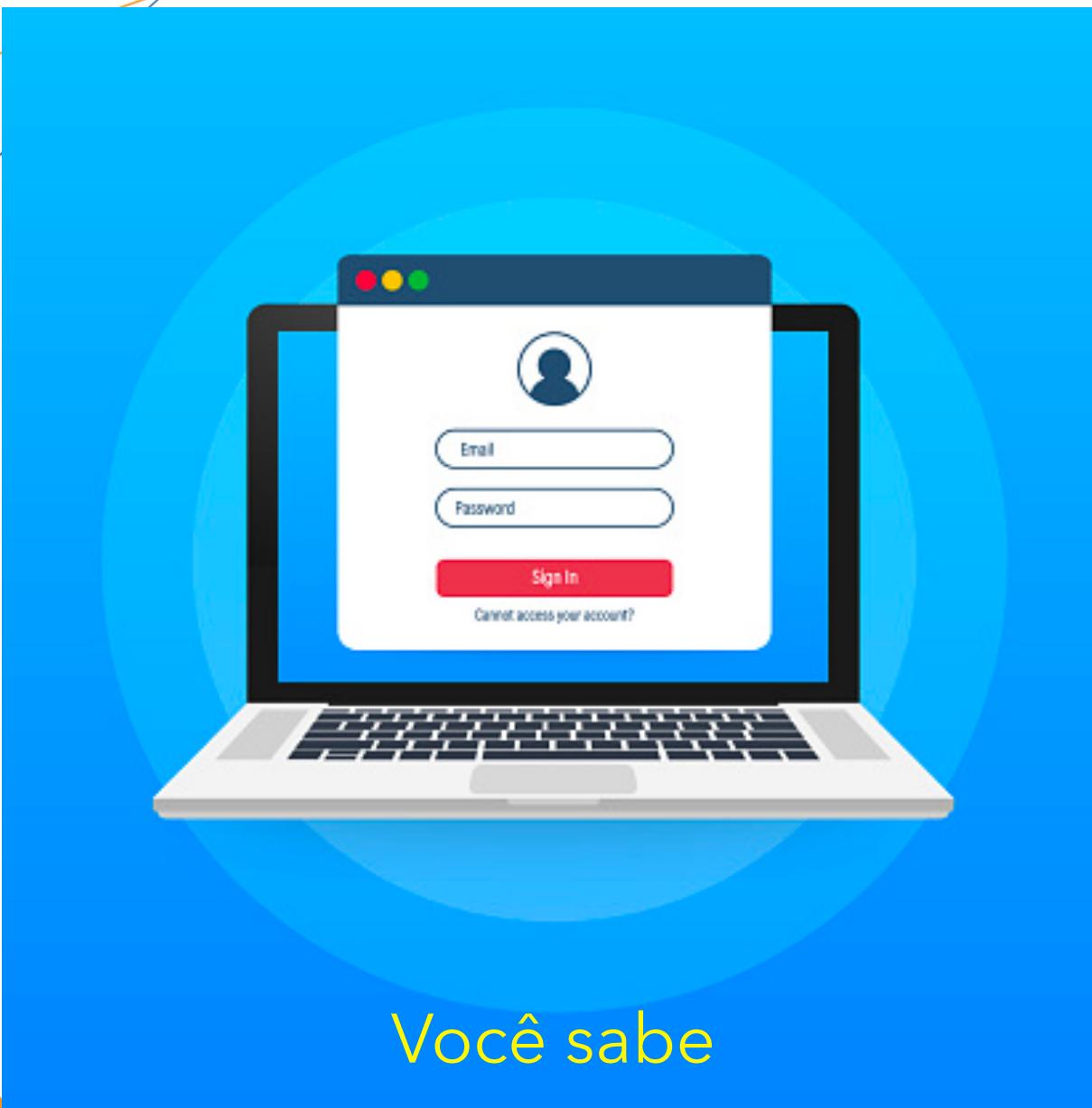
Bases Tecnológicas, científicas e instrumentais (conteúdos):

- Processo de Autenticação e Biometria;

Situação de Aprendizagem:

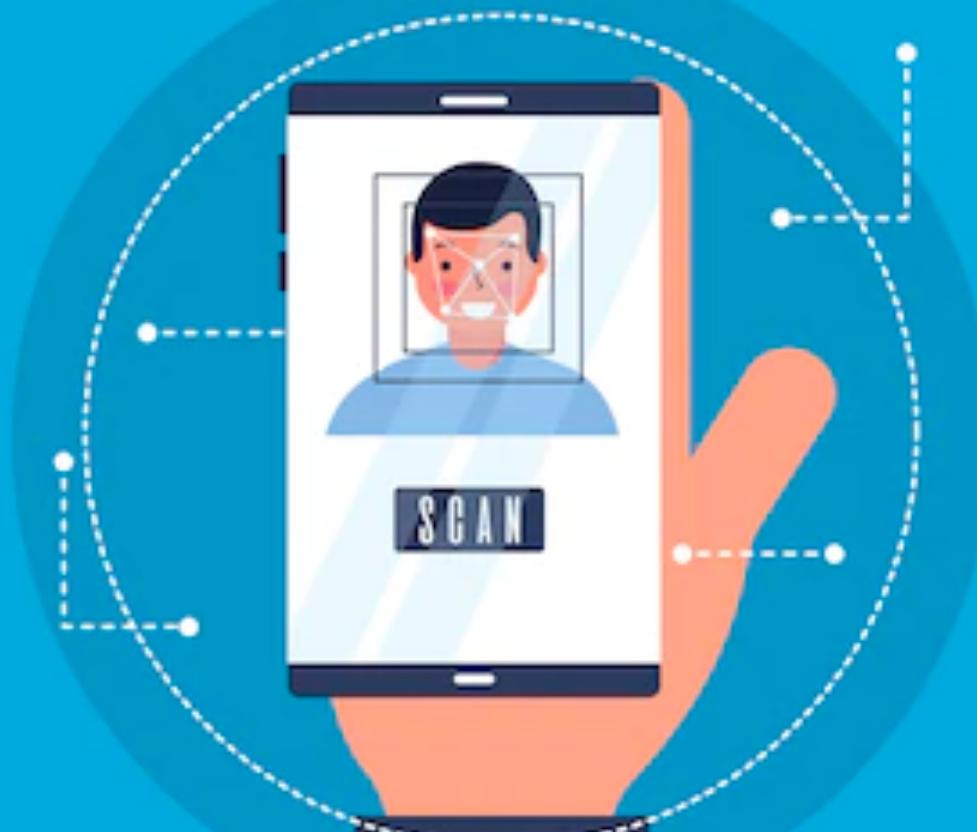
- Assinatura e Certificação Digital.

Autenticação



Você sabe

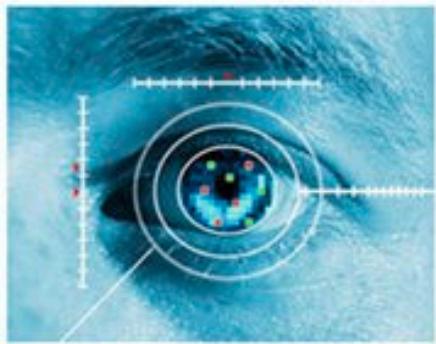
Biometria



Você tem

Tipos de Biometria

Physiological



Iris



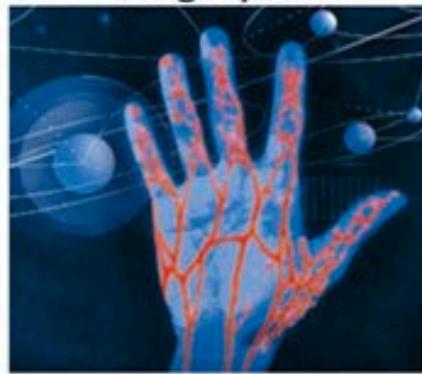
Fingerprint



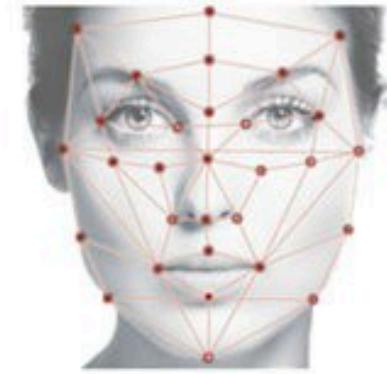
Ear



DNA

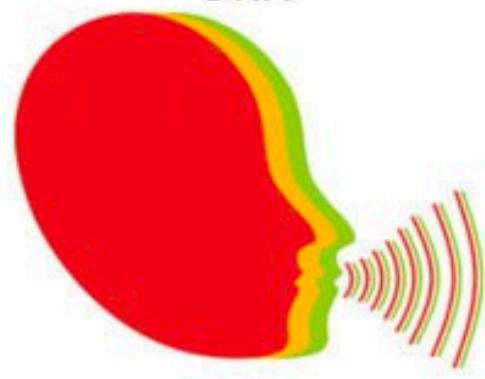


Vein print



Face

Behavioral



Voice



Gait



Signature

ac

Verificação e Identificação

Verificação

- 1 para 1 – você declara quem é, e o sistema compara as características apresentadas com a previamente adquirida

Identificação

- 1 para n – o sistema faz uma varredura de um conjunto de características e decide se uma delas é da pessoa identificada

FAR e FRR

A precisão de um sistema biométrico pode ser categorizada por duas medidas:

- Taxa de falsa aceitação (FAR – False Acceptance Rate)
- Taxa de falsa rejeição (FRR – False Rejection Rate)

FAR e FRR

A FAR representa a percentagem de usuários **não-autorizados** que são **incorrectamente identificados** como usuários válidos.

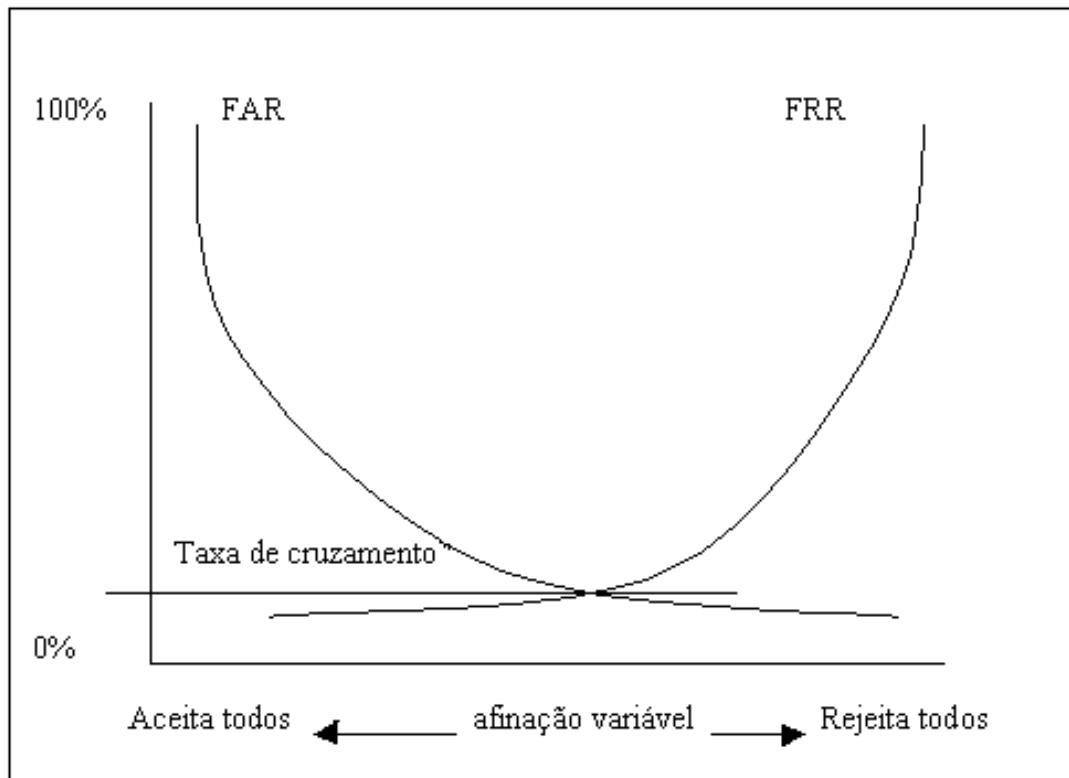
A FRR representa a percentagem de usuários **autorizados** que são **incorrectamente rejeitados**.

A falsa rejeição pode causar frustrações.

A falsa aceitação pode permitir **fraudes**.

FAR e FRR

O ponto onde a FRR e a FAR são equivalentes é denominado Taxa de Erro de Cruzamento (Crossover Error Rate – CER)



Tipos de Sistemas de Autenticação

Baseado em características físicas

- Impressão Digital
- Geometria da mão
- Reconhecimento da face
- Identificação da íris
- Identificação da retina

Tipos de Sistemas de Autenticação

Baseado em características comportamentais

- Reconhecimento da voz
- Reconhecimento da assinatura
- Dinâmica da digitação.

Requerimentos de um Sistema Biometrico

Teoricamente, qualquer característica humana, física ou comportamental, pode ser usada para a identificação de pessoas, desde que satisfaça os seguintes requerimentos:

Universalidade: significa que todas as pessoas devem possuir a característica;

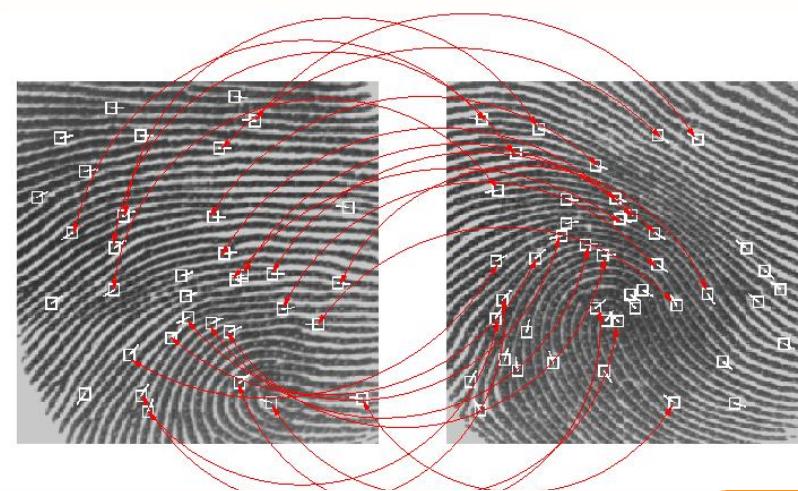
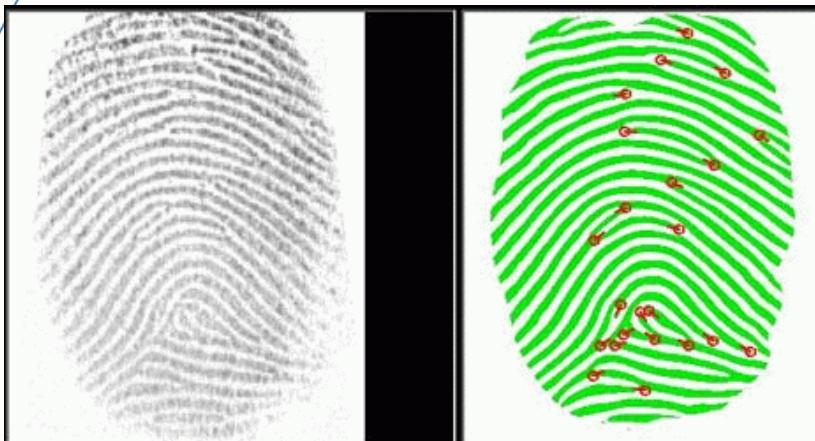
Singularidade: indica que esta característica não pode ser igual em pessoas diferentes;

Permanência: significa que a característica não deve variar com o tempo;

Mensurabilidade: indica que a característica pode ser medida quantitativamente.

Impressão Digital

Na verificação de uma impressão, muitos sistemas analisam a posição de detalhes chamados de minutiae (minutas).



Impressão Digital

Sistemas modernos também verificam outras características para identificação única, tais como:

- Arcos;
- Voltas;
- Delta;

Por exemplo:

Alguns dispositivos contam o número de cumes entre um minutiae para formar o modelo de referência.

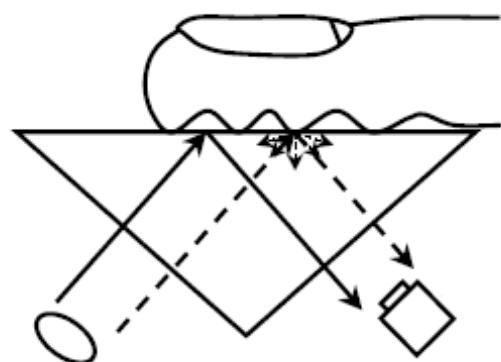
Impressão Digital

Sensores:

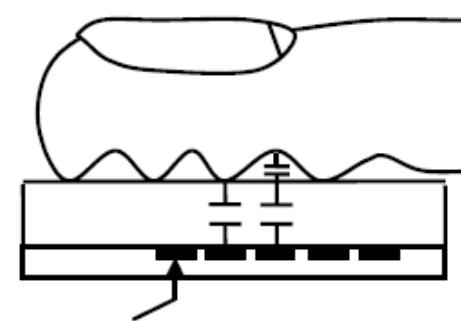


Impressão Digital

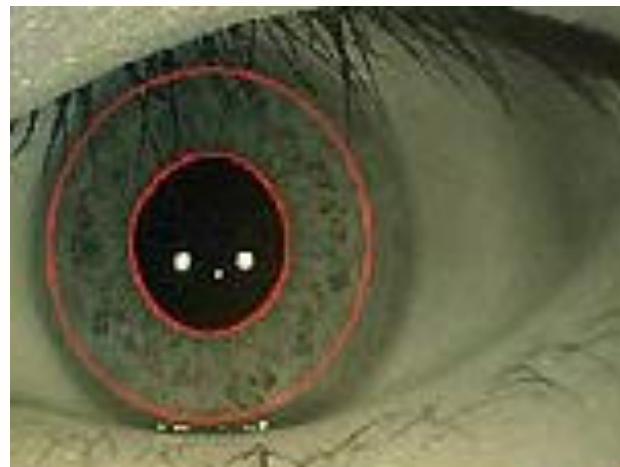
Sensores do tipo óptico:



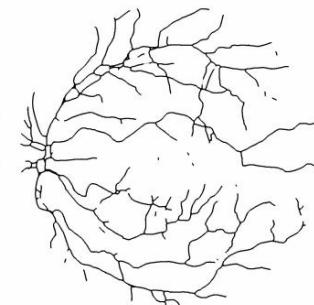
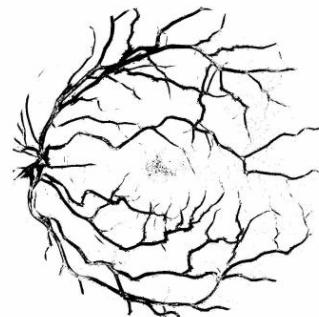
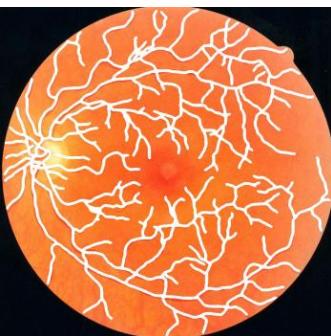
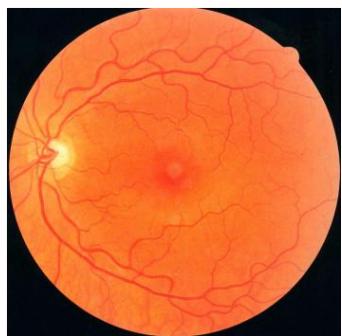
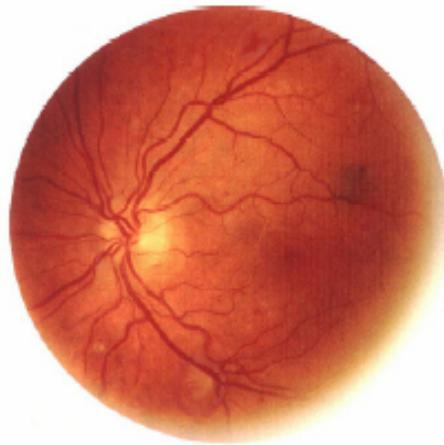
Sensores do tipo capacitivo:



Íris

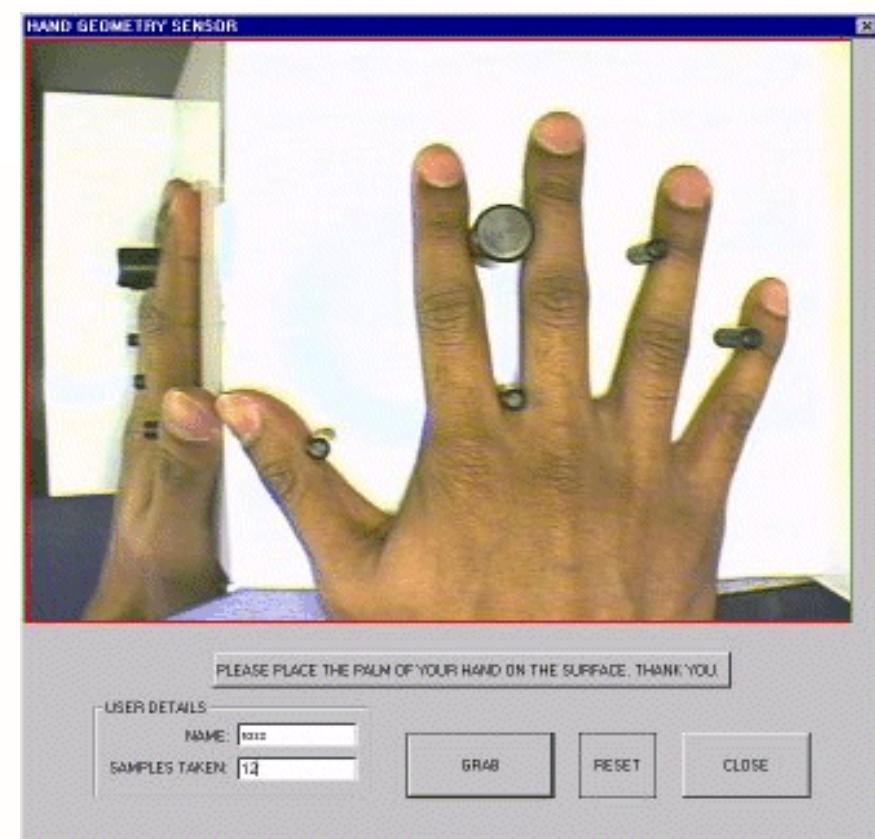


Retina



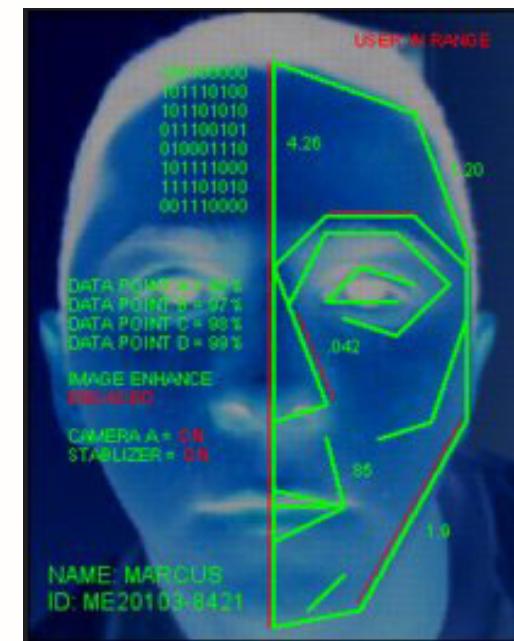
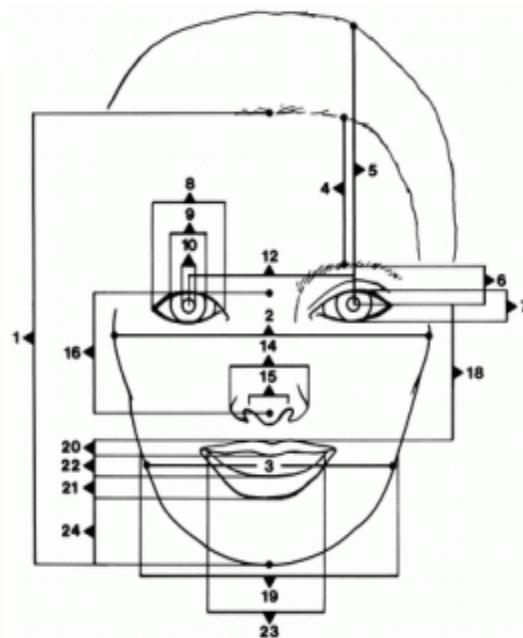
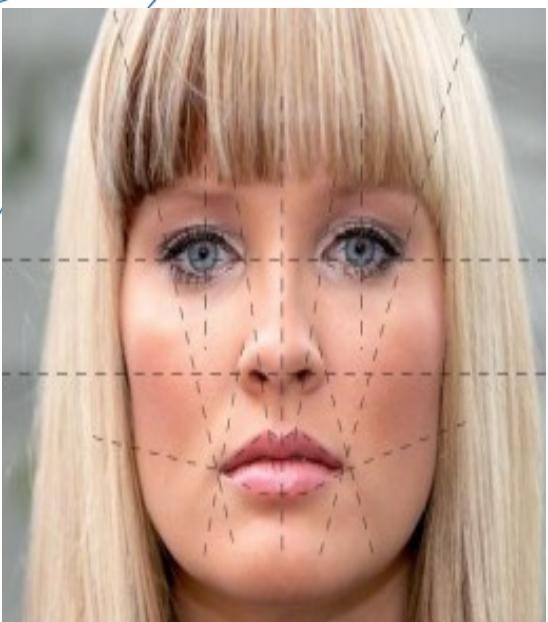
senac

Geometria da mão

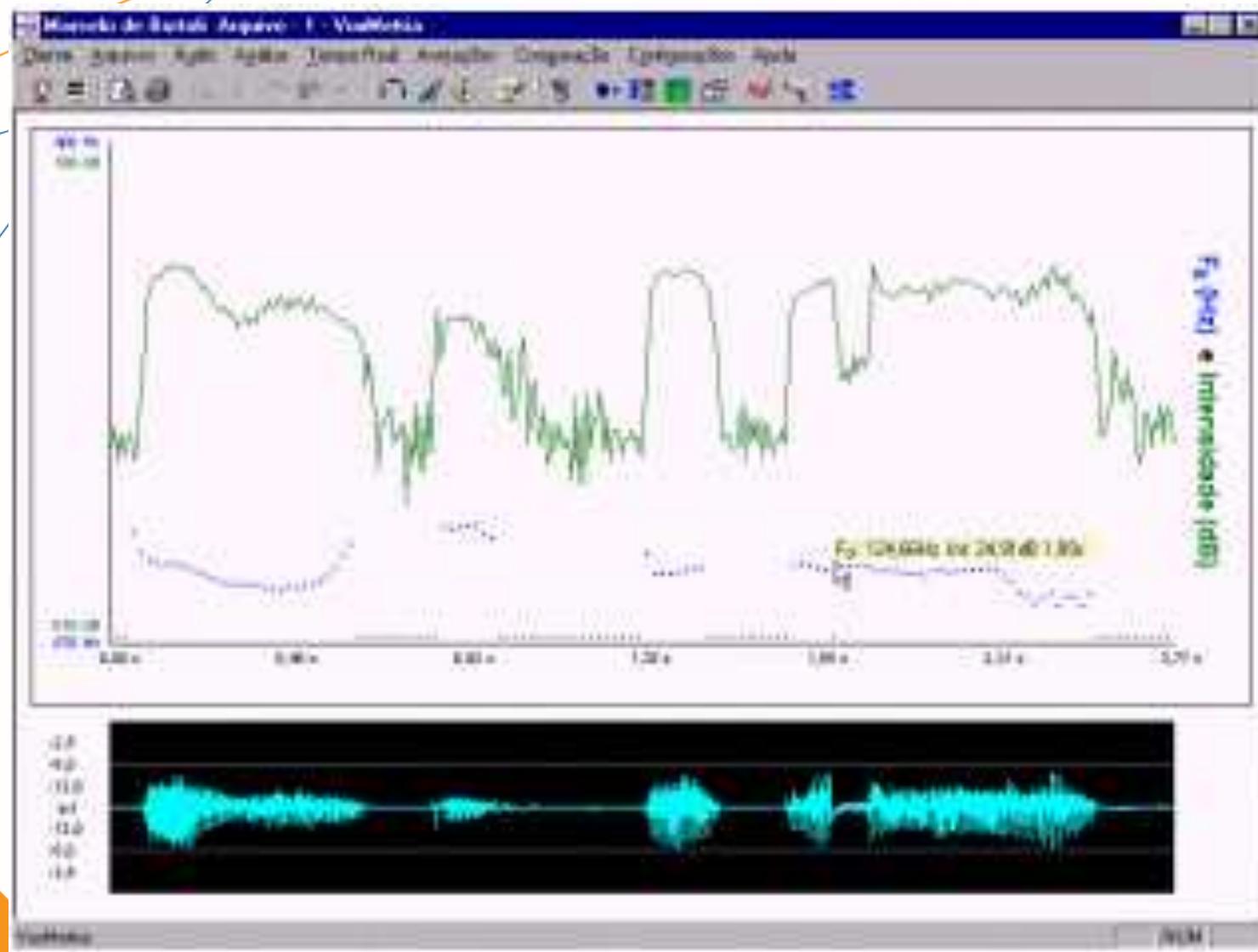


Senac

Reconhecimento de face



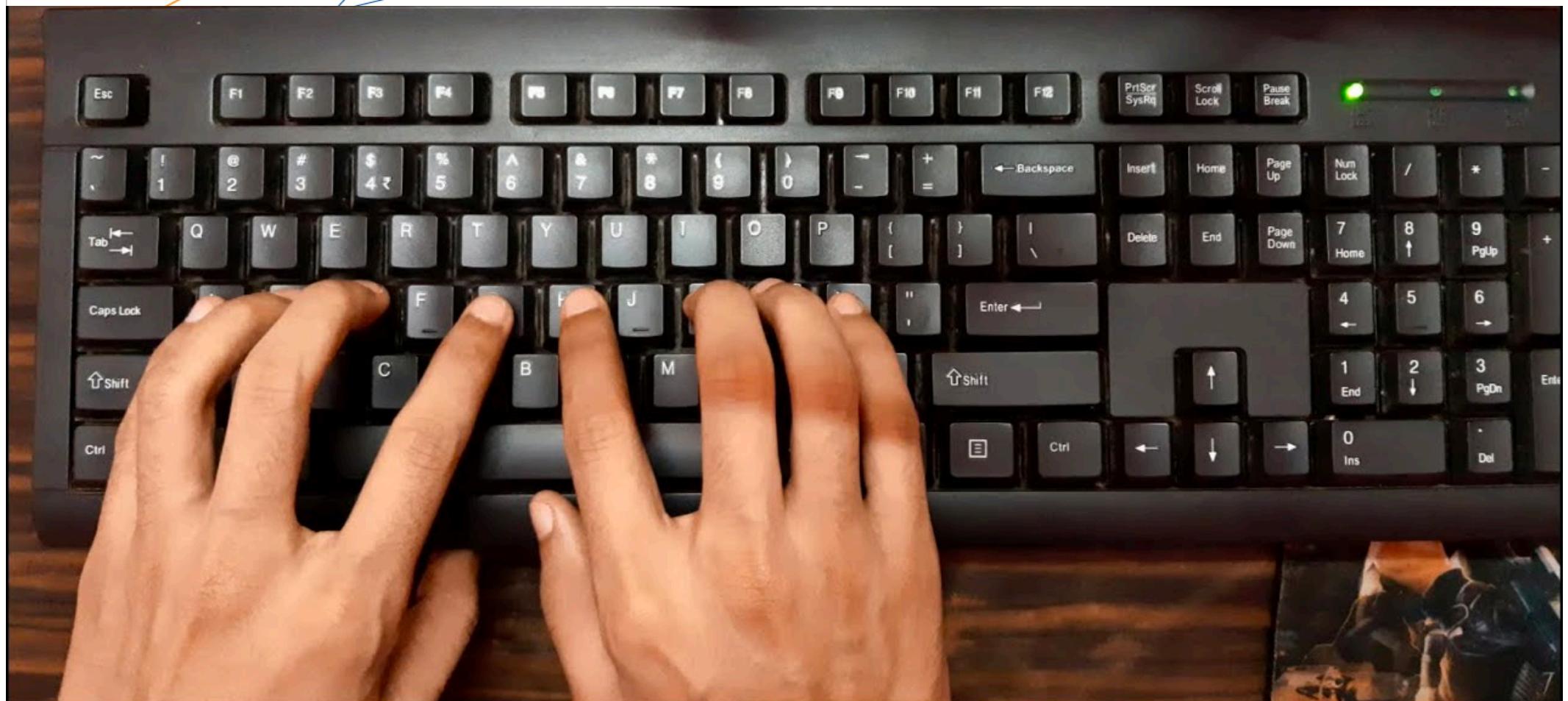
Reconhecimento da Voz



Reconhecimento da Assinatura



Dinâmica da Digitação



Exemplo Prático

```
import cv2
from google.colab.patches import cv2_imshow
import numpy as np
import os

test_original = cv2.imread("finger-print.png")
cv2_imshow(test_original)
cv2.waitKey(0)
cv2.destroyAllWindows()

match_points = []
for file in [file for file in os.listdir("./database")]:
    print(file)
    fingerprint_database_image = cv2.imread("./database/"+file)

    sift = cv2.xfeatures2d.SIFT_create()

    keypoints_1, descriptors_1 = sift.detectAndCompute(test_original, None)
    print(keypoints_1)
    keypoints_2, descriptors_2 = sift.detectAndCompute(fingerprint_database_image, None)
    print(keypoints_2)
    matches = cv2.FlannBasedMatcher(dict(algorithm=1, trees=10),
    dict()).knnMatch(descriptors_1, descriptors_2, k=2)

    for p, q in matches:
        if p.distance < 0.1*q.distance:
            match_points.append(p)
```

Exemplo Prático

```
keypoints = 0
if len(keypoints_1) <= len(keypoints_2):
    keypoints = len(keypoints_1)
else:
    keypoints = len(keypoints_2)
print(keypoints)
print(len(match_points))
if (len(match_points) / keypoints)>0.3:
    print("% match: ", len(match_points) / keypoints * 100)
    print("Fingerprint ID: " + str(file))
    result = cv2.drawMatches(test_original, keypoints_1, fingerprint_database_image,
keypoints_2, match_points, None)
    result = cv2.resize(result, None, fx=2.5, fy=2.5)
    cv2_imshow(result)
    cv2.waitKey(0)
```