

# **SISTEMA DE CONTROLE DE ACESSO COM TRÊS FATORES DE AUTENTICAÇÃO PONTO DE CONTROLE 2**

*Mateus Felipe Massa, Victor Bastos I. Oliveira*

Programa de Graduação em Engenharia Eletrônica, Faculdade Gama  
Universidade de Brasília  
Gama, DF, Brasil  
email: mateusmassa7@gmail.com, victorbio101@gmail.com

## **1. DESCRIÇÃO DE HARDWARE**

Na montagem inicial do sistema foram utilizados três principais hardwares, a fim de realizar uma prova de conceito para o projeto final. A placa Raspberry Pi 3 modelo B foi utilizada como unidade de processamento central, responsável por controlar os periféricos de acordo com o código embarcado, além de executar a primeira parte do controle de acesso. Para compilar e executar os arquivos/códigos necessários na placa foi utilizado acesso remoto através do programa VNC Viewer.

Conectado aos pinos da Raspberry Pi, o módulo RFID-RC522 foi utilizado como periférico para validação de acesso, agindo como o segundo fator em um sistema com três fatores de autenticação. Este módulo foi interfaceado com a placa através do protocolo de comunicação SPI, ocupando um total de cinco pinos, desconsiderando alimentação e aterramento. Duas tags padrão MIFARE1 S50, uma do tipo chaveiro e outra do tipo cartão, foram utilizadas para testar a funcionalidade do módulo.

Em uma das portas USB da placa foi conectado uma webcam Logitech C270, de resolução máxima de 720p em 30 FPS. Este periférico foi utilizado para a checagem do último fator de au-

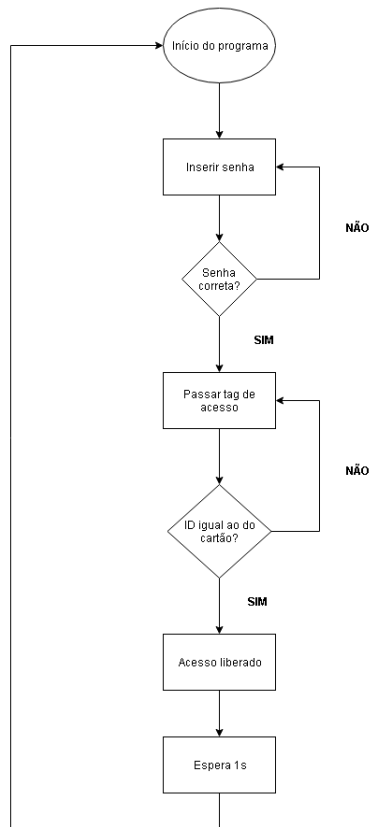
tenticação, dado através do reconhecimento facial do usuário.

## **2. DESCRIÇÃO DE SOFTWARE**

O software embarcado no sistema foi constituído por dois blocos principais: um bloco contendo os dois primeiros fatores de autenticação (senha e tag RFID) e o segundo bloco com a autenticação por meio do reconhecimento facial.

No primeiro bloco foi desenvolvido um código na linguagem C/C++ no qual foi feita a lógica da validação da senha a partir da comparação da string inserida pelo usuário com uma string preestabelecida que contém a senha salva. Caso as strings forem iguais o usuário era direcionado para a validação da tag de acesso, caso contrário, o programa informava o erro e pedia a senha novamente. O restante do código contém a parte do sistema RFID para a validação da tag de acesso, no qual foi tomado como base um exemplo da biblioteca MRFC522, desenvolvida para a comunicação do módulo RFID-RC522 com a Raspberry Pi [1]. A partir da função que recolhe o ID da tag lida, foi implementado uma lógica que concatena esses dados em uma string e compara com strings que contém o ID do cartão e do chaveiro. Caso a string formada tivesse um ID igual a do

cartão, o usuário teria o acesso liberado, caso contrário o programa esperaria para realizar outra leitura. Uma visão geral do primeiro bloco pode ser obtida através do fluxograma abaixo, que explica como ocorre o fluxo de informações do código:

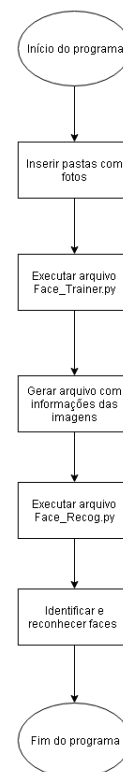


**Fig. 1.** Fluxograma de validação de senha e RFID

O segundo bloco consiste em dois códigos na linguagem Python que foram utilizados no teste sem serem modificados e são divididos em: treinamento do código e execução do reconhecimento facial [2]. Os códigos foram desenvolvidos com apoio das bibliotecas OpenCV (uma biblioteca desenvolvida na linguagem C/C++ mas que tem suporte para a linguagem Python) e a biblioteca dlib, que foi desenvolvida na linguagem C/C++ mas é multiplataforma [3]. Ambas as bibliotecas foram desenvolvidas para atuarem em sistemas de detecção e reconhecimento facial, possuindo funções especializadas nestes assuntos.

Inicialmente, para a utilização do código foi

necessário incluir duas pastas com imagens do rosto das pessoas a serem reconhecidas, uma pasta por pessoa com o nome de cada pessoa em cada pasta. Após isso executou-se o código de treinamento, que habilitou o programa de reconhecimento para reconhecer os rostos compatíveis com os das fotos passadas na fase de treinamento. Esse programa gerou um arquivo que contém as informações sobre os arquivos de fotos e que traduz essas informações em vetores de valores que a máquina entende. O próximo passo foi executar o programa de reconhecimento facial, que com o apoio da webcam detectou os rostos em seu campo de visão, reconhecendo os rostos caso eles fossem iguais aos rostos cadastrados. Para representar a sequência de passos necessários para a execução do programa, criou-se o fluxograma abaixo:



**Fig. 2.** Fluxograma programa de reconhecimento facial

### 3. RESULTADOS PRELIMINARES

Com a configuração de hardware inicial e códigos não totalmente integrados, os resultados obtidos foram satisfatórios como prova de conceito para o objetivo final. A autenticação do primeiro fator, uma senha numérica qualquer, foi testado e validado com sucesso já na linguagem C/C++. Ao entrar com uma senha diferente da estabelecida em código, o programa imprimiu na tela que a senha estava incorreta e exigiu uma nova inserção de senha. Ao acertar a senha o programa passou para o próximo estágio de autenticação. O sistema RFID também foi testado e validado com sucesso utilizando C/C++. Ao passar a tag do tipo chaveiro o acesso foi negado, porém ao passar a tag do tipo cartão o acesso foi liberado e o programa retornou a solicitar a senha, como previsto.

Nos testes com a webcam para o reconhecimento facial foram encontradas diversas dificuldades. A detecção de faces funcionou inteiramente, tanto presencialmente quanto para fotos mostradas para a câmera. Já o reconhecimento facial em si demonstrou um comportamento fora do previsto, funcionando parcialmente apenas para fotos mostradas para a câmera. Ao tentar reconhecer ao vivo um dos integrantes do grupo, que estavam cadastrados no banco de dados, o programa não acusava a identidade da pessoa. Ao colocar uma foto para a webcam analisar, o integrante era identificado, mas sua identidade acusada não era consistentemente a correta.

### 4. REFERENCIAS

- [1] P. Thorell, “Mfrc522.” [Online]. Available: <https://github.com/paguz/RPi-RFID>
- [2] A. Raj, “Real time face recognition with raspberry pi and opencv.” [Online]. Available: <https://circuitdigest.com/microcontroller-projects/raspberry-pi-and-opencv-based-face-recognition-system>
- [3] D. E. King, “dlib c++ library.” [Online]. Available: <https://github.com/davisking/dlib>