16º Congresso de Inovação, Ciência e Tecnologia do IFSP - 2025

REIMPLEMENTAÇÃO DA BIBLIOTECA DE TRATAMENTO DE IMPRESSÕES DIGITAIS SOURCEAFIS EM *C++*

JOÃO P. GEBARA[1], LUÍS H. SACCHI[2] RAFAEL A ABREU[3].

[1] Cursando Bacharelado em Ciência da computação, IFSP Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, Campus Salto joao.gebara@aluno.ifsp.edu.br.
[2] Professor do Instituto Federal de São Paulo, IFSP Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, Campus Salto sacchi@ifsp.edu.br
[3] Cursando Bacharelado em Ciência da computação, IFSP Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, Campus Salto abreu.rafael@aluno.ifsp.edu.br

Área de conhecimento (Tabela CNPq): 1.03.03.02-2 Engenharia de Software

RESUMO: O projeto objetiva realizar uma implementação em *C++* de uma biblioteca de identificação de impressões digitais SourceAFIS do autor Važan (2023), originalmente escrita em *Java* e *C# .NET*. Almeja-se melhorar o desempenho geral da biblioteca para poder ser usada em sistemas embarcados nos quais os recursos computacionais são limitados. O algoritmo de reconhecimento de impressão digital é capaz de comparar duas impressões digitais ao estabelecer o grau de similaridade entre duas leituras do mesmo dedo, realizadas com um sensor biométrico.

PALAVRAS-CHAVE: Otimização; Biometria; Linguagem *C++*; sistemas embarcados;

REIMPLEMENTATION OF THE SOURCEAFIS FINGERPRINTING LIBRARY IN *C++*

ABSTRACT: The project objective was to implement in *C++* the digital wealth interpretation library SourceAFIS, written by Važan (2023), originally written in *Java* and *C# .NET*. The aim is to improve the library's overall performance so that it can be used in embedded systems where computing resources are limited. The fingerprint recognition algorithm can compare two fingerprint generations using an algorithm that establishes the degree of similarity between two readings of the same finger, taken with a biometric sensor.

KEYWORDS: Optimization; Biometrics; *C++* language; Programming.

INTROUÇÃO

Este projeto foi pensado como apoio a um projeto de extensão do IFSP *campus salto* (2025), o projeto de descaracterização de TV Box. Os estudantes do IFSP desinstalam o software malicioso da TV Box, que é usado para acesso irregular a canais desbloqueados de TV e instalam o sistema operacional Linux, transformando os equipamentos em computadores de baixo custo.

A biblioteca SourceAFIS foi originalmente escrita em linguagem *Java* e *C# .NET*, linguagens que dão resultados insatisfatórios no quesito de otimização. Este projeto almeja facilitar acesso à essa tecnologia em máquinas de baixo desempenho, o objetivo principal é converter toda a biblioteca SourceAFIS para a linguagem *C++* podendo ser compilada para arquiteturas x64, x86 e ARM, buscando um desempenho superior à sua versão original em *Java* e *.NET*.

Temos o objetivo de utilizar o programa nessas máquinas de baixo custo, facilitando e democratizando o acesso ao sistema de leitura de digital para várias finalidades, como por exemplo, seria útil em salas de aula, podendo agilizar o processo de chamada com um método mais assertivo,

diferente dos métodos tradicionais, como o oral e escrito.

MATERIAL E METÓDOS

Por conta de ser um software de computador, será necessário apenas os computadores pessoais para o desenvolvimento, e o próprio sistema embarcado de leitura de impressão digital, mas no quesito softwares serão utilizados alguns como, por exemplo:

QT Creator com GCC: IDE que será usada para construção do software em *C++* com o compilador Gnu GCC embutido na plataforma.

Testes unitários: Para os testes unitários será utilizado o Google test.

Intellij Idea: Escolhido para a depuração do programa em *Java*.

GCC: Depurador de Compiladores de C e *C++*.

Git e Github Actions: Para automatizar o desenvolvimento do trabalho e organizar as tarefas demandadas

Visual Studio: Testes multiplataforma no windows e utilização da ferramenta de Debug.

Dataset: O repertório de digitais está disponível no programa do autor.

Sobre método de desenvolvimento, um adequado para a conversão da biblioteca é o método ágil, com o uso em conjunto do quadro kanban, organizando assim os pontos principais do programa e desenvolvendo em partes, sendo vital sempre manter a comunicação do que tem de ser feito, está sendo feito e finalizado.

O dispositivo embarcado alvo tem as seguintes especificações:

TABELA 1. Configuração da máquina de baixo desempenho

Tipo	Especificação
Processador	Amlogic S905X4 ARM Cortex A55, Quad Core
GPU	Mali G31 MP2
Memória RAM	2 GB
Tipo de memória	DDR4
Espaço de armazenamento	16 GB
Tipo de armazenamento	eMMC

Seu sistema operacional é baseado em Linux, no momento a distro Armbian está sendo usada, isso está suscetível a mudanças.

No momento o compilador está configurado para o ambiente de desenvolvimento, as seguintes tags estão presentes no arquivo de build:

TABELA 2. Flags de compilação

Flag	Descrição
/DWIN32	Define macro WIN32 (Windows 32-bit)
/D_WINDOWS	Define macro _WINDOWS (aplicação Windows)
/MDd	Runtime library debug multithreaded DLL
/Ob0	Desabilita inline expansion
/Od	Desabilita otimização (modo debug)
/RTC1	Runtime checks básicas (stack frame, variables)
-std:*C++*17	Padrão *C++*17 (projeto principal)
-std:*C++*20	Padrão *C++*20 (sourceafis-cpp)
/ZI	Informação de debug Edit and continue
/GS	Buffer security check
/W4	Nível de warning 4 (avisos altos)
/WX	Trata warnings como erros
/wd4251	Desabilita warning 4251 (interface DLL)
/wd4275	Desabilita warning 4275 (interface DLL não exportada)

/nologo	Suprime banner do compilador
/J	Muda char padrão para unsigned
/D_UNICODE	Define Unicode
/DUNICODE	Define Unicode
/DSTRICT	Define strict type checking
/DWIN32_LEAN_AND_MEAN	Exclui APIs menos usadas do Windows
/wd4702	Desabilita warning 4702 (código inalcançável)
/utf-8	Codificação fonte UTF-8
/DGTEST_HAS_PTHREAD=0	GoogleTest sem suporte a pthread
/D_HAS_EXCEPTIONS=1	Habilita exceções
/machine:x64	Arquitetura alvo x64
/debug	Gera informações de debug
/INCREMENTAL	Linking incremental
/subsystem:console	Subsistema console para aplicação

Por conta de o projeto estar na fase de desenvolvimento, foi desativado as flags de otimização para melhor proveito das ferramentas de debug.

RESULTADOS E DISCUSSÃO

Como explica Marcondes (2020), a biometria é a autenticação ou identificação de um indivíduo pelas suas características biológicas físicas e comportamentais, o nosso corpo tem várias características que podem nos diferenciar de outro, umas das mais conhecidas são as impressões digitais dos nossos dedos.

O método de reconhecer uma pessoa pela sua digital é muito mais antigo do que se imagina, relatos históricos apontam que já se usava essa tecnologia antes do nascimento de Cristo, Paoli et. al (2016) explicam que por meados do século XIX, já eram pesquisados padrões nas digitais. Foi Francis Galton quem criou o método que usamos nos dias de hoje, dando o nome de minúcias para as características únicas presentes nas digitais de cada pessoa. Essas minúcias podem ser classificadas como fim de linha, bifurcações e entre outras, conforme a Figura 1. A biblioteca SourceAFIS identifica as minúcias para comparar duas impressões digitais.

FIGURA 1. Exemplos de minúcias

Com base nesses pontos é feita a comparação, utilizando a biometria, que estuda as características físicas com o intuito de diferenciá-las, e assim, identificá-las, um processo que está sendo muito utilizado, como é afirmado por Luciano R, et al (2006) a biometria vem se popularizando, sendo apontada como uma solução consolidada, segura e barata para problemas de autenticação.

A biblioteca original possui um total de 132 classes em *Java* onde 22 são testes unitários, os quais priorizamos a conversão e aproximadamente 90% já foi implementado. Todos os recursos para testes e auxiliares já foram implementados, assim como o ambiente *C++* que exige recursos, ferramentas e configurações diferentes do projeto principal.

CONCLUSÕES

A implementação da biblioteca SourceAFIS em *C++* se demonstra viável pela similaridade com as outras linguagens em que foi implementada e pelo fato do dispositivo embarcado apresentar melhor desempenho ao lidar com um executável compilado em relação a um programa que precisa do ambiente *Java* ou *.NET* (*C#*).

A implementação original em *Java* se mostra menos eficiente, além de ter apresentado conflitos com o sistema embarcado usado.

A versão em *C#* se mostra superior em relação à biblioteca original, entretanto levantou preocupações referentes ao ambiente de produção e desenvolvimento, já que apesar de multiplataforma, historicamente tem um foco e suporte melhor para Windows, e a plataforma de testes e produção usará uma distribuição baseada em Linux com arquitetura ARM.

O porte para *C++* se mostrou eficiente em suprir as deficiências de ambas as versões, sendo executada de forma nativa no ambiente Linux e tendo uma performance acima da original. Entretanto, a conversão feita teve foco em uma rápida conversão. Sendo assim, muitas técnicas de otimização da linguagem *C++* ainda podem ser aplicadas para aumentar a performance da aplicação.

CONTRIBUIÇÕES DOS AUTORES

R.A.A e J.P.G. contribuíram para a programação e desenvolvimento do software, com J.P.G. também trabalhando na configuração de ambiente. R.A.A. desenvolveu os primeiros documentos, enquanto J.P.G. revisava e corrigia. L.H.S. idealizou o projeto, cedeu os equipamentos de hardware necessários e orientou o desenvolvimento por todas as etapas. Os envolvidos revisaram e aprovaram a versão submetida.

REFERÊNCIAS

IFSP Campus Salto, **Projeto TVBox Descaracterização e catalogação,** 23 maio 2025. Acesso em 1 ago. 2025. Online. Disponível em: https://www.youtube.com/watch?v=T8nNGwOOCaY

LUCIANO R. et al**. Introdução à Biometria. Departamento de Automação e Sistemas.** UFSC, 2006.

MARCONDES, José Sérgio**. Biometria, Sistema Biométrico: O que é, Como Funciona?** 2020.

PAOLI, Antonio; DAHIA, Gabriel; AMPARO**,** Lucas**. Uso de impressão digital para biometria**, 2016.

VAŽAN, Robert. **SourceAFIS fingerprint matcher**. 2023. Acesso em 21 maio. 2025. Online. Disponível em: https://sourceafis.machinezoo.com