DESENVOLVIMENTO DE APLICATIVO CELULAR PARA ACESSO A EDIFÍCIOS VIA SENHAS INDIVIDUAIS, RECONHECIMENTO FACIAL E RECONHECIMENTO BIOMÉTRICO

Domênica dos Santos
Fernanda Moralez
Mayara Santos Chinaglia
Sérgio Luis Rabelo de Almeida
Edvaldo Angelo
Silvia Maria Stortini González Velázquez
Jorge Alexandre Onoda Pessanha
Universidade Presbiteriana Mackenzie (UPM)

Resumo

Este artigo tem como objetivo desenvolver uma solução tecnológica para permitir a entrada de pessoas autorizadas em condomínios e residências, evitando, assim, impasses durante os períodos noturnos e nas madrugadas, visto que algumas pessoas retornam de suas atividades nesse período e se deparam com um atraso para adentrarem suas residências, o que as deixam sujeitas a roubos e furtos. Além disso, há a questão da crise sanitária vivenciada agora em relação à pandemia ocasionada pela Covid-19, em que áreas e superfícies comum se tornaram alvo de grande preocupação quanto à contaminação pelo vírus. A solução apresentada foi a criação de um aplicativo como método de

entrada em residências ou condomínios, que funciona por meio de leitura biométrica e reconhecimento facial no próprio celular do indivíduo. Esse aplicativo foi desenvolvido através do *framework Flutter* conectado a uma fechadura elétrica e um microcontrolador (NodeMCU), proporcionando uma autenticação individual da entrada de cada condômino para adentrar sua residência. Os resultados mostraram que o sistema desenvolvido teve um desempenho satisfatório com tempo de conexão entre o dispositivo e a fechadura inferior a dois segundos, além de promover aumento de higiene ao adentrar as residências, haja vista o uso individual do aplicativo via *smartphone*.

Palavras-chave: Residências. Aplicativo. Segurança biométrica.

1 INTRODUÇÃO

Vivenciamos um momento histórico no qual diversas redes de conexões são criadas diariamente e são cada vez mais comuns na vida das pessoas, por meio da utilização de *smartphones*, *notebooks*, *smartTVs* ou outros dispositivos que alteram os relacionamentos interpessoais e o modo como o ser humano vive e interage com o mundo ao seu redor.

Com o desenvolvimento de dispositivos eletrônicos móveis inteligentes em 2007, introduzidos pela Apple, e em 2008 pela Google, desencadeou-se a necessidade de criar programas para ampliar as funcionalidades dos *smartphones*. Com isso, surgiram os aplicativos que, atualmente, estão presentes na vida dos indivíduos de diversos modos, por apresentarem inúmeras funcionalidades, como acessar a conta bancária sem precisar ir a uma agência, solicitar e utilizar transporte individual, pedir refeições, comunicar-se e, até mesmo, fazer acompanhamento de saúde e qualidade de vida (LIMA, 2019).

O mercado de desenvolvimento de *apps* vem evoluindo vertiginosamente na atualidade, devido ao aumento ascendente do uso de aplicativos no cotidiano das pessoas e, sobretudo, pelo crescimento acelerado das tecnologias desenvolvidas. Segundo a consultoria App Annie, o Brasil é o terceiro colocado no *ranking* dos países em termos de tempo gasto em *apps*, com um aumento de 40% no número de *downloads* de aplicativos realizados por ano desde 2017 (VALENTE, 2020).

Com os avanços na elaboração de aplicativos, o modo de programar voltou-se ao uso facilitado e desenvolvimento de interfaces mais intuitivas e simplificadas, de maneira que não é essencial ter grande conhecimento em programação para criar bons

programas. Uma pessoa que possua certo grau de entendimento sobre o assunto é capaz de usufruir, atualmente, das ferramentas que antes eram utilizadas com um grau de complexidade maior.

No âmbito de desenvolvimento de aplicativos, três vertentes de desenvolvimento destacam-se e lideram o mercado atual: *web*, nativo e híbrido. Muitas métricas são consideradas no momento da escolha do desenvolvimento com que se trabalhará, como a linguagem de programação, a manutenção, a comunidade e a curva de aprendizagem, "que medirá as dificuldades encontradas durante o desenvolvimento com cada abordagem. De forma objetiva, ela é uma representação do nível médio cognitivo de aprendizagem para uma determinada atividade ou ferramenta" (REIS, 2019, p. 25), entre outros.

Cada sistema operacional tem suas particularidades para o desenvolvimento dos aplicativos. Os aplicativos em iOS podem ser desenvolvidos apenas em computadores Mac que rodam o Mac OS X, com auxílio de um único ambiente de desenvolvimento integrado (IDE), o XCode, além de realizar o *download* de um *kit* de desenvolvimento de *software* (SDK). Os aplicativos em Android possuem mais flexibilidade de desenvolvimento, uma vez que podem ser criados em diversos sistemas operacionais, como Windows, Linux, inclusive o Mac OS X, e possuem vasta possibilidade de escolha de IDEs, como Android Studio e Intellij IDEA, assim como a necessidade de *downloads* de SDKs.

Os *frameworks* proporcionaram grandes avanços ao desenvolvimento de aplicativos devido à capacidade de serem *Cross-Platform* (multiplataforma). Essa característica viabiliza a redução do custo de produção e manutenção, já que utiliza linguagens de programação geralmente empregadas em aplicações *Web*, como JavaScript. Segundo Bristowe (2017 *apud* LIMA, 2019, p. 36), "o código-fonte é encapsulado em uma aplicação nativa e utiliza-se de WebView ou componentes de web para apresentar a interface, acessar bibliotecas internas do SO e para comunicar-se com os sensores embutidos no aparelho". Os *frameworks* mais utilizados são: *Ionic*, *React Native* e *Flutter*.

Adicionalmente ao *framework*, torna-se necessária a utilização de um banco de dados para o armazenamento das informações. Até o presente momento, existem diversos tipos de bancos de dados e, dependendo das necessidades do usuário, basta selecionar o mais adequado. Os bancos de dados são classificados em: Bancos de dados relacionais; Bancos de dados orientado ao objeto; Bancos de dados distribuídos; Data *warehouse*, Bancos de dados NoSQL; Bancos de dados de gráfico; Bancos de dados OLTP; Bancos de dados de código aberto; Bancos de dados em nuvem; Bancos de dados multimodelo; Bancos de dados de documentos/JSON; e Bancos de dados autônomos, conforme contextualiza a Oracle Brasil (S. d.).

Também é importante contextualizar o *hardware* a ser utilizado, que necessariamente passa pelo emprego dos microcontroladores, o qual apresentou um crescimento

exponencial a partir do instante em que foram notadas as diversas aplicabilidades existentes dentro da área industrial. Ressalta-se que, desde a década de 1970, o ramo dos sistemas embarcados ascendeu de forma bastante acentuada. Grandes empresas começaram a investir nessa esfera promissora da tecnologia, visto que isso proporcionaria a possiblidade de executar automaticamente as tarefas que antes eram realizadas de forma manual. Esses sistemas ganharam novos componentes, oferecendo uma série de novas funcionalidades para o mercado e ganhando o nome de microcontroladores (OLIVEIRA, 2017).

Outro aspecto relevante diz respeito às ferramentas de reconhecimento singular do indivíduo, que têm sido cada vez mais instauradas nos novos modelos tecnológicos de acesso. O uso de características biológicas humanas não só apura a segurança, reduzindo os riscos de violações ocorridas pela identificação incorreta da pessoa que solicita acesso a espaços privados, mas também oferece vantagens adicionais para o usuário, como a conveniência de não ter nenhum cartão de acesso físico ou mesmo senhas para lembrar (WANG *et al.*, 2015).

Essa implementação tem se mostrado bastante presente, desde o acesso a grandes edifícios até a concessão de acesso aos dados de pequenos dispositivos celulares. Entre essas ferramentas, há o reconhecimento biométrico, também conhecido por biometria, cuja história começou nos últimos 50 anos, sendo a primeira publicação científica datada de 1963 (OTTI, 2017).

Entende-se que "Reconhecimento biométrico, ou simplesmente, biometria é a ciência que estabelece a identidade de uma pessoa baseada em suas características físicas ou comportamentais, assim como impressão digital, facial, íris e voz" (JAIN; ROSS; NANDAKUMAR, 2011, p. 7, tradução nossa). Tais formas distintas de atribuir a identificação do indivíduo, de modo geral, podem ser classificadas em dois grupos. O primeiro grupo é baseado nas características fisiológicas do sujeito, isto é, traços advindos da carga genética, dificilmente transformados no decorrer do tempo. Já o segundo grupo, direcionado ao reconhecimento biométrico, baseia-se em características que são adquiridas ou desenvolvidas com o passar do tempo; assim, essas características oscilam ao decorrer da vida humana. Não obstante, dependendo do estado atual e do desejo de cada sujeito, esses atributos podem oscilar de maneira acentuada, essas peculiaridades podem ser notadas no tom de voz e na assinatura (COSTA; OBELHEIRO; FRAGA, 2006).

É possível citar diversas abordagens na literatura dos últimos anos que reforçam o interesse da comunidade científica pelo tema, incluindo novas formas de identificação, como segurança das informações. Rabinezhadsadatmahaleh e Khatibi (2020) elaboraram um algoritmo com base em modelos de aprendizado de máquina convencionais e profundos para a identificação humana a partir de sinais de eletrocardiograma (ECG). Sabe-se que sinais de ECG mostram indicativos únicos de características

comportamentais das pessoas devido à morfologia e à estrutura do coração que as tornam mais apropriado para a identificação humana.

Já Wang *et al*. (2015) desenvolveram um dispositivo para a aquisição de imagem infravermelha da veia dorsal da mão, que se forma objetivando obter padrões e métodos de processamento de imagem eficientes de computação para a identificação biométrica. Os resultados foram promissores, com características de operação para uso no mundo real que incluem confiabilidade, facilidade de uso, alta segurança, rapidez na resposta, baixo custo e simples instalação.

Por sua vez, Hu *et al*. (2018) investigaram como terceirizar o processo de varredura de alto custo computacional mantendo a privacidade do banco de dados e da computação em um ou dois servidores dedicados, visto que a identificação biométrica normalmente verifica um banco de dados de grande escala de registros biométricos para encontrar uma correspondência próxima o suficiente de um indivíduo. Complementando, Zhu *et al*. (2018) propuseram um protocolo para terceirização de informações biométricas com uso de computação em nuvem, realizando operações de identificação sobre o banco de dados criptografado e retornando o resultado ao proprietário do banco de dados. Uma análise preliminar indicou que o esquema proposto por Zhu *et al*. (2018) é seguro e com melhor desempenho, mesmo que os invasores tentem forjar solicitações de identificação por meio da nuvem.

Adicionalmente, tem-se observado um aumento no uso da identificação facial devido à sua natureza menos intrusiva, mas devemos ressaltar que o seu desempenho de reconhecimento é sensível às condições da imagem, como mudanças de iluminação, oclusões parciais por óculos ou cabelo, diferentes ângulos de pose e variações de expressões faciais (WANG *et al*., 2015).

Diante dos argumentos apresentados, este artigo objetiva relatar o desenvolvimento de um sistema de ingresso automático direcionado a residências e condomínios, baseado em reconhecimento facial e biométrico, e capaz de ser executado a partir de um *smartphone*, com segurança e rapidez, em contrapartida aos sistemas de porteiro eletrônico, em que a decisão de acesso é feita por uma pessoa, muitas vezes a distância, com ocorrências de atraso e falhas de identificação.

2 METODOLOGIA

Os procedimentos adotados para desenvolvimento deste projeto utilizaram um modelo padrão de biblioteca disponível na plataforma de programação. A primeira

etapa foi o estabelecimento dos requisitos para a escolha do *software* que contemplaram o desempenho na abertura da fechadura (redução do tempo de abertura), simplicidade de uso, plataforma de uso baseado em *smartphones* e segurança, seja do ponto de vista sanitário, seja do ponto de vista de acesso.

Para este projeto, selecionou-se o desenvolvimento híbrido devido à facilidade de entendimento para programadores iniciantes e à gama de bibliotecas já prontas aliado ao IDE Android Studio; além de servir como ferramenta voltada para a criação de códigos, há um emulador incorporado a ele, o que torna possível executar o código criado e testá-lo em um simulador de *smartphone Android* (emulador) (ANDROID STUDIO, 2020).

Após pesquisas e análises feitas sobre esses três *frameworks*, foi selecionado o *Flutter*, tendo em vista o seu desenvolvimento baseado em *widgets*, que representam todas as camadas que estarão presentes na interface do usuário final. Sua linguagem de programação diferencia-se dos outros *frameworks*, pois utiliza uma linguagem própria; o Dart, entretanto, é bem semelhante às demais. Por não utilizar outras plataformas, esse *framework* promete entregar boa *performance* aliada à grande facilidade de programação, criando uma curva de aprendizagem ainda mais baixa. Por ser um *framework* novo no mercado, sua comunidade ainda não é tão expressiva quanto à dos *frameworks* já citados, entretanto o *Flutter* apresenta melhor desempenho em relação às demais por utilizar uma linguagem de programação avançada AOT (*Ahead-Of-Time*) e não necessitar de um *framework* para modificar o código para nativo (FLUTTER).

O banco de dados selecionado foi o *Firebase*, por sua facilidade de integração com o *Flutter*, ambos desenvolvidos pela Google. O *Firebase* é caracterizado por um banco de dados em tempo real, que fornece uma API que permite aos desenvolvedores armazenar e sincronizar dados entre vários clientes. A maioria dos seus recursos são gratuitos, para qualquer nível de desenvolvimento, e, quando um aplicativo desenvolvido com uso desse banco de dados demanda maior armazenamento de dados, não é necessário escalar o código ou mudar o servidor para obter capacidade de armazenamento extra, mas tão somente adquirir um plano comercial de maior capacidade (FIREBASE).

Para o desenvolvimento deste trabalho foi selecionado o microcontrolador NodeMCU, haja vista que é muito utilizado por oferecer boa usabilidade aliada ao baixo custo, à variedade de ambientes de desenvolvimento e à disponibilidade de módulos periféricos. Ele apresenta mais benefícios em relação ao Arduino, em termos de processamento, frequência de relógio, quantidade de memória e variedade de interfaces de comunicação (OLIVEIRA, 2017). Essas placas de desenvolvimento incluem o ESP8266 já soldado, com pinos para encaixar em uma *protoboard*, memória *flash* externa e conversor serial-USB. Com isso, pode ser ligado ao PC com um simples cabo USB e o carregamento de programas pode ser feito sem maiores complicações.

Uma de suas grandes vantagens em relação a outros microcontroladores é a presença da interface *Wi-Fi*. Não é necessário acoplar nenhum módulo extra para utilizar esse tipo de interface de comunicação. Essa característica faz do ESP8266 um grande candidato para aplicações voltadas ao IoT. Esse microcontrolador apresenta a possibilidade de programação em três ambientes de desenvolvimento diferentes, entre eles: um ambiente baseado na linguagem Lua (linguagem de alto nível e fácil integração); a própria IDE do *Arduino*; e um ambiente baseado no sistema operacional de tempo real (RTOS), em que é possível aproveitar o total potencial do microcontrolador (OLIVEIRA, 2017).

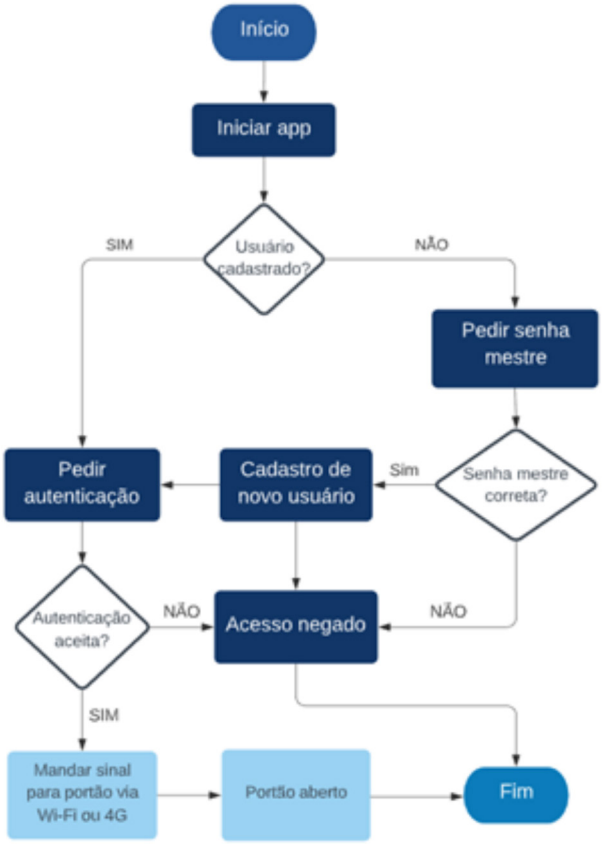

Figura 1 Fluxograma do funcionamento do *app*.

Fonte: Elaborada pelos autores.

No desenvolvimento desse aplicativo, devido à pré-existência da biometria digital e o reconhecimento facial nos *smartphones*, optou-se por implementar as duas formas de autenticação, deixando a critério do usuário o modo biométrico que será utilizado.

Dessa forma, o aplicativo foi elaborado por meio do *framework Flutter*, o qual é apropriado para o desenvolvimento híbrido de *apps*. As informações geradas pelo aplicativo são armazenadas através da conexão via 4G ou *Wi-Fi* do celular com o banco de dados (*Firebase*), e a integração dessas informações e o acionamento da fechadura é feito por meio do microcontrolador (NodeMCU). Na Figura 1, há um fluxograma simples sobre o funcionamento do aplicativo com as atividades de execução do *software*.

2.1 Conexão com o banco de dados

Antes de iniciar a integração das funcionalidades eletrônicas e do *framework*, é necessário realizar a configuração do banco de dados, pois este servirá como um elo para a comunicação entre o *app* e o microcontrolador. No banco de dados, é armazenado o estado atual da fechadura (aberta ou fechada) e o controle das pessoas que estarão autenticadas a utilizar o *app* para oferecer uma camada de segurança extra. Nesse projeto, para permitir a utilização do aplicativo, foi elaborada a autenticação em dois fatores: a inserção de uma senha mestre e um *login*, constituído por *e-mail* e senha.

Para essa configuração, é necessário entrar no *site* do Firebase, que é um banco de dados da Google, e criar um banco de dados para uma conta que ficará responsável pelo gerenciamento dos dados ali presentes. Com o banco de dados criado, a autenticação por *e-mail* e senha deve ser habilitada e uma divisão em formato JSON deve ser adicionada para armazenar o estado dos portões e uma senha mestre (Figura 2).

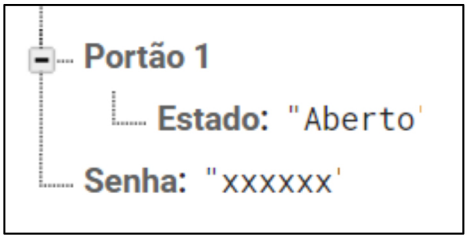

Figura 2 Configuração dos dados no formato JSON.

Fonte: Elaborada pelos autores.

Feitas essas modificações, o aplicativo deve ser adicionado e configurado ao *Firebase* para que haja comunicação entre ele e o banco de dados de forma segura. O *app* pode ser configurado para sistemas operacionais *Android* e *IOS*. Com essas alterações, o banco de dados está apto para dar continuidade ao desenvolvimento do projeto.

2.2 Desenvolvimento com microcontrolador

A seção eletrônica, responsável pela conexão entre o aplicativo e a fechadura do portão, é desenvolvida usando-se um módulo relé e um módulo NodeMCU, que conta com um microcontrolador Esp8266 e um módulo de comunicação *Wi-Fi*.

O módulo de comunicação recebe as informações do banco de dados e transporta-as para que o microcontrolador interprete o comando em que o módulo relé deve ser acionado ou não. O NodeMCU, por sua vez, possui uma programação própria que receberá o sinal do módulo *Wi-Fi* e o interpretará, para que o módulo relé seja acionado por meio das devidas conexões. O módulo relé possui o papel de simular a fechadura de um portão, que, por conseguinte, poderá ser conectado a uma fechadura por suportar tensões usuais presentes em componentes residenciais.

2.2.1 Integração microcontrolador com módulo relé

A comunicação entre o microcontrolador e o módulo relé é executada por meio de uma programação na interface IDE Arduino e um simples circuito elétrico para que as portas lógicas do NodeMCU estejam conectadas ao módulo relé e o sinal possa ser levado de um módulo ao outro. Na programação, será definida uma GPIO que levará o sinal para o relé ser acionado e, ainda em teste, o circuito eletrônico será montado em uma placa *protoboard*, que conecta os componentes de um circuito real de forma prática e simples.

Na Figura 3, pode-se observar a montagem do circuito eletrônico para que o teste da programação seja realizado e o relé possa ser acionado. O NodeMCU é alimentado via conexão USB entre o *laptop* e a sua porta de entrada, que também permite realizar a compilação do código desenvolvido para o microcontrolador. O acionamento pode ser percebido pelo *LED* aceso no módulo relé e no próprio NodeMCU.

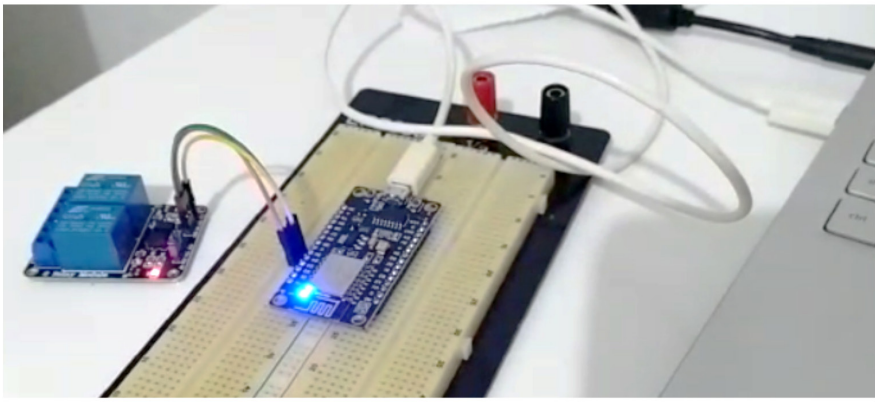

Figura 3 Simulação do circuito eletrônico para o módulo relé.

Fonte: Acervo pessoal dos autores.

2.2.2 Integração microcontrolador com banco de dados

A integração entre o banco de dados e o NodeMCU é feita mediante uma programação com uma biblioteca específica, criada para a interface Arduino IDE, com o intuito de interpretar bancos de dados do Firebase.

Nessa programação, é informada a rede *Wi-Fi* em que o NodeMCU ficará conectado constantemente e um código de segurança, o qual apenas o administrador do banco de dados possui acesso, reforçando a segurança na troca de dados. Através dessas conexões, o NodeMCU recebe os dados em formato JSON e os interpreta.

Para o projeto em questão, o NodeMCU procura o estado do portão em formato JSON e aciona o relé, caso esteja indicado como aberto, e desativa o relé caso o estado esteja indicado como fechado, como demonstra o fluxograma da Figura 4:

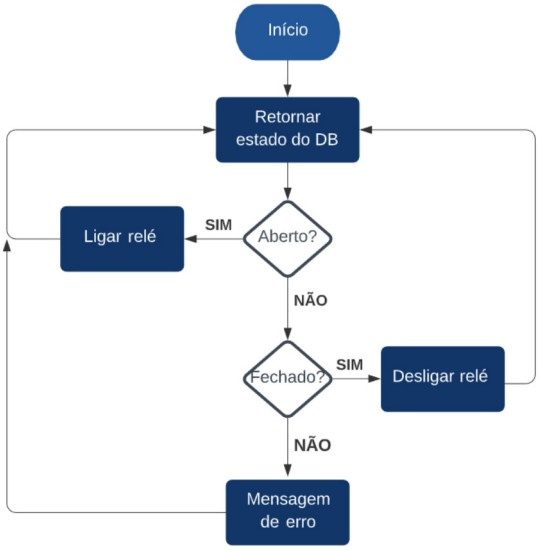

Figura 4 Lógica aplicada para o acionamento do relé.

Fonte: Elaborada pelos autores.

2.3 Desenvolvimento do aplicativo

Para o desenvolvimento do aplicativo, duas ferramentas foram consideradas: a linguagem de programação e o *framework* associado. No tocante à programação e à criação dos códigos, utilizou-se a linguagem Dart, a qual está acoplada ao uso do *framework Flutter*.

Ao escolher a linguagem de programação e o *framework*, o próximo passo foi a seleção do ambiente de desenvolvimento do código da aplicação, o qual, nesse projeto, foi o Android Studio.

2.3.1 Integração do aplicativo com banco de dados

A integração do aplicativo com o banco de dados é demasiadamente importante por ser a parte responsável pela inserção do estado atual do portão (aberto ou fechado).

Como primeiro teste para a inserção de dados, no momento da programação, foi criado um botão responsável por abrir e fechar o portão (Figura 5), para que, ao

pressioná-lo, um sinal do tipo "aberto" fosse enviado ao banco de dados e, após alguns segundos, o sinal do tipo "fechado".

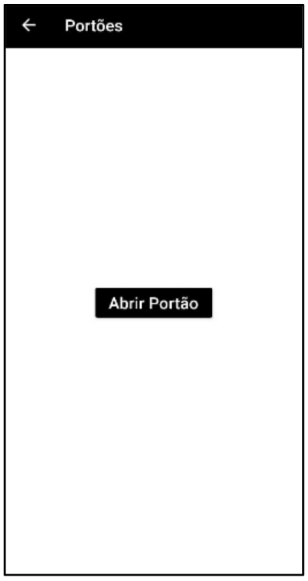

Figura 5 Tela do *app* responsável pela abertura dos portões.

Fonte: Elaborada pelos autores.

2.3.2 Autenticação por senha mestre

Para que haja mais segurança e gerenciamento dos usuários que terão acesso ao aplicativo, realiza-se uma verificação com uma senha mestre, fazendo com que o usuário não possa prosseguir com o registro antes dessa primeira etapa, logo, a inserção da senha mestre só é necessária quando o usuário se registra. Essa senha será fornecida apenas aos usuários que realmente devem ter acesso ao *app* e só poderá ser alterada pelos gerenciadores do banco de dados, como explicado no tópico 2.1.

Como pode ser observado na tela inicial (Figura 6), há uma notificação, abaixo do botão de registro, que indica o estado da autenticação da senha mestre. Para autenticar, é necessário pressionar o botão "Autenticar" e digitar a senha mestre na tela que surgirá: se a senha inserida estiver correta, a tela inicial será exibida novamente com a notificação que o usuário está autenticado. Realizada essa etapa, o usuário pode seguir com o registro normalmente.

Figura 6 Telas para a autenticação da senha mestre.

Fonte: Elaborada pelos autores.

2.3.3 Autenticação por e-mail *e senha*

Como forma de *login* no aplicativo foi escolhida a autenticação por *e-mail* e senha, dessa forma, é possível gerenciar os usuários logados por meio do banco de dados. Essa autenticação é feita na primeira tela de interação com o *app*, em que o usuário pode escolher entre registrar um *e-mail* e senha ou logar, caso já tenha se registrado, como mostra as figuras 6 e 7:

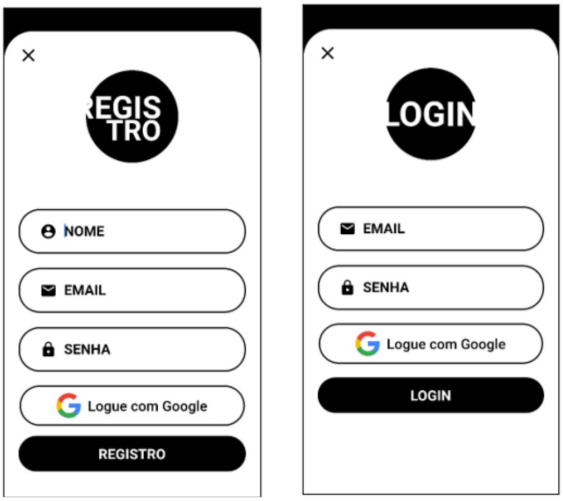

Figura 7 Telas de registro e *login* do *app*.

Fonte: Elaborada pelos autores.

Ao logar ou se registrar, o usuário será levado à tela principal do *app* (Figura 8), em que será mostrado o nome do usuário cadastrado e um botão que dará acesso à tela dos portões (Figura 5):

Figura 8 Tela principal do *app*.

Fonte: Elaborada pelos autores.

2.3.4 Autenticação por biometria digital e reconhecimento facial

Após a elaboração e integração das configurações citadas anteriormente, mais uma camada de segurança é adicionada: o reconhecimento via impressão digital. A partir dela, é possível garantir um nível de proteção maior para as residências, conforme será abordado no tópico da revisão de literatura referente à biometria.

A autenticação por biometria digital será solicitada toda vez que o usuário desejar entrar na tela responsável pela abertura dos portões, na sequência, a impressão digital será requerida quando o usuário estiver em sua tela principal (Figura 8) e pressionar o botão "Portões". Ao pressionar o botão, uma nova janela será aberta, solicitando a digital, a verificação é feita através da digital já cadastrada no próprio dispositivo, fazendo com que só o proprietário do aparelho tenha controle sobre o uso e o acesso às funcionalidades do aplicativo. Caso a digital verificada esteja correta, o usuário é levado à tela que possibilita a abertura dos portões (Figura 9).

É importante salientar que a mesma biblioteca utilizada para a programação da biometria do dispositivo oferece também a possibilidade de utilizar o reconhecimento facial, dado que ambas as ferramentas são da própria tecnologia de autenticação do celular, sendo assim, tanto a biometria quanto o reconhecimento facial podem ser utilizados, como sinalizadas na Figura 9. Vale pontuar que, no entanto, nem todos os dispositivos celulares possuem o reconhecimento facial, impossibilitando o seu uso no aplicativo.

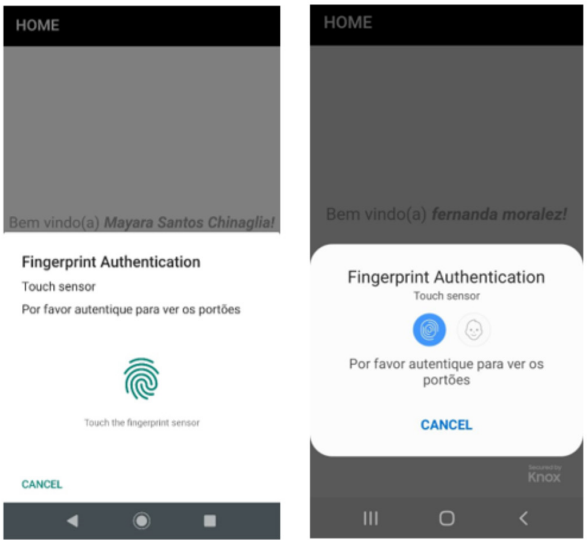

Figura 9 Autenticação por biometria digital ou reconhecimento facial.

Fonte: Elaborada pelos autores.

2.4 Desenvolvimento do protótipo físico

Após a análise de todos os testes e a validação da correta atuação da seção eletrônica e das funcionalidades do *app*, faz-se necessário o início do desenvolvimento do protótipo.

2.4.1 Prototipagem da placa

Com a parte eletrônica finalizada, o circuito montado na *protoboard* pode ser transferido para uma placa de circuito impresso, também conhecida como PCB (*Printed Circuit Board*). São placas destinadas ao uso de circuitos eletrônicos por possuírem uma camada de materiais condutores que ficarão responsáveis pelas conduções elétricas no circuito. Há várias formas de confeccionar uma PCB e muitos *softwares* já estão disponíveis para facilitar a fabricação delas e tornar o resultado mais profissional. No entanto, há uma série de materiais que precisam ser adquiridos para que a placa possa ser confeccionada, além da necessidade de um local adequado, já que o processo inclui o uso de produtos químicos para a corrosão do material metálico.

Para que essa dificuldade seja contornada, em caso de circuitos simples, uma ótima opção é o uso de placas universais perfuradas de fenolite. Essa placa já conta com o material condutor em um dos seus lados e já é perfurada, facilitando muito a confecção do circuito, pois só será necessário soldar os componentes eletrônicos em suas devidas conexões.

Nesse projeto, uma placa de fenolite perfurada foi escolhida para confeccionar o circuito entre o NodeMCU e o módulo relé, que, posteriormente, será conectado à fechadura eletrônica. Na Figura 10 há o resultado da fabricação do circuito eletrônico:

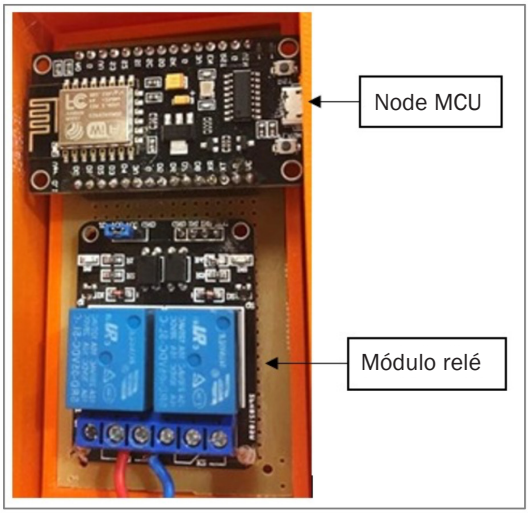

Figura 10 Circuito impresso desenvolvido para o protótipo.

Fonte: Elaborada pelos autores.

2.4.2 Montagem da parte elétrica

A parte elétrica será responsável por receber o sinal do relé e acionar a fechadura eletrônica. O módulo relé pode ser ligado a um transformador que possui uma entrada para botões auxiliares, a qual converterá o sinal do relé de 5V para 12V, e estará conectado à fechadura eletrônica ou a um interfone, que possui uma entrada para botões auxiliares, que, por sua vez, estará conectado à fechadura eletrônica. Nesse projeto, foi escolhida a ligação entre o módulo relé e o interfone por oferecer um local mais protegido para o circuito eletrônico e um sinal de *Wi-Fi* mais estável. O sinal proveniente do interfone será levado à fechadura eletrônica e, por conseguinte, acionará a abertura do portão. A Figura 11 exemplifica a ligação entre os componentes elétricos e eletrônicos:

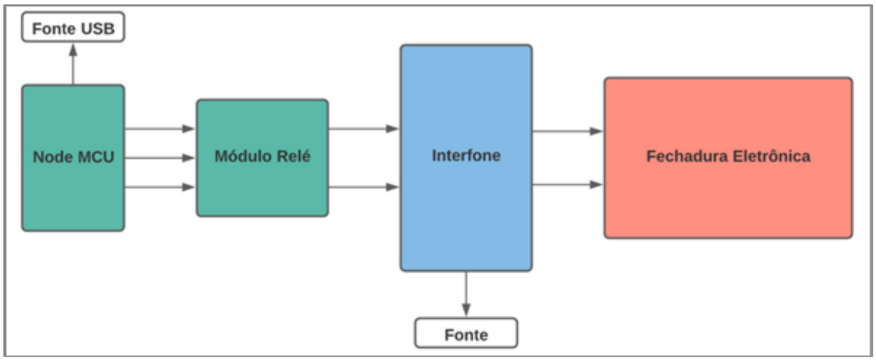

Figura 11 Esquema de ligação entre os componentes elétricos e eletrônicos.

Fonte: Elaborada pelos autores.

2.4.3 Montagem da caixa de proteção

Para fins de segurança do circuito eletrônico, uma caixa foi projetada através do *software Inventor* e produzida por impressão 3D, com o intuito de armazenar a placa contendo o NodeMCU e o módulo relé, oferecendo proteção contra fatores externos. A Figura 12 mostra a caixa produzida:

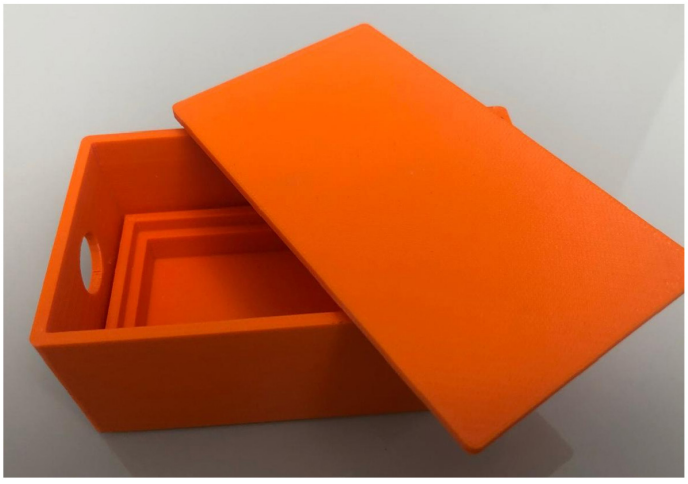

Figura 12 Caixa de proteção produzida por impressão 3D.

Fonte: Elaborada pelos autores.

2.4.4 Protótipo final

Após a avaliação de todos os componentes elaborados, a união de todas as peças pode ser feita e o protótipo testado. A Figura 13 mostra o protótipo físico desenvolvido:

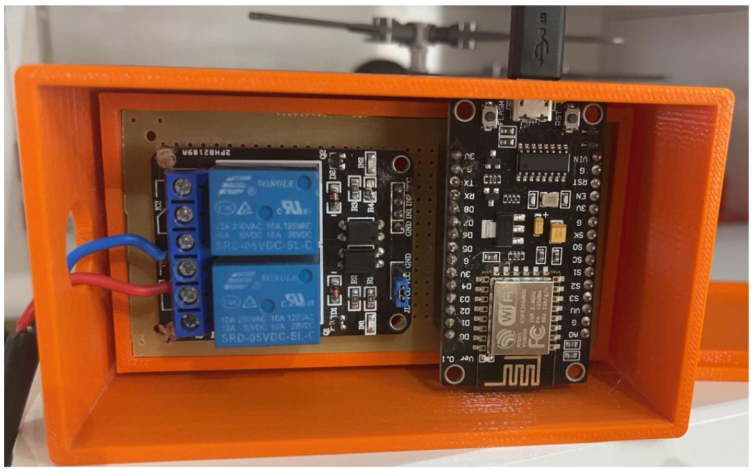

Figura 13 Protótipo final desenvolvido.

Fonte: Acervo pessoal dos autores.

A partir do protótipo, a eficiência do projeto foi medida por meio de duas métricas:

a) tempo médio de acionamento da fechadura eletrônica pelo aplicativo;
b) comparação da eficiência dos tempos médios consumidos para abertura do portão através de uma chave e do aplicativo.

A métrica "a" pode ser vista na Figura 14:

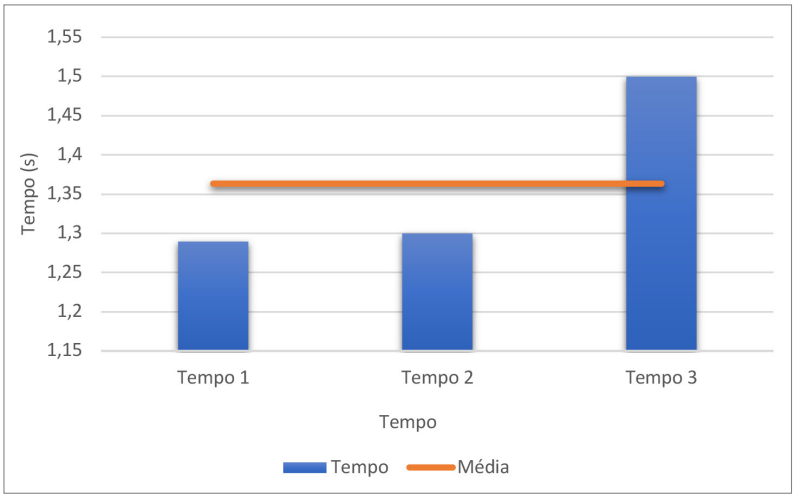

Figura 14 Gráfico de tempos médios de acionamento da fechadura eletrônica pelo do aplicativo.

Fonte: Elaborada pelos autores.

Os tempos de acesso foram medidos com cronômetro e realizados a partir de conexão 4G do celular.

A métrica "b" pode ser observada na Figura 15:

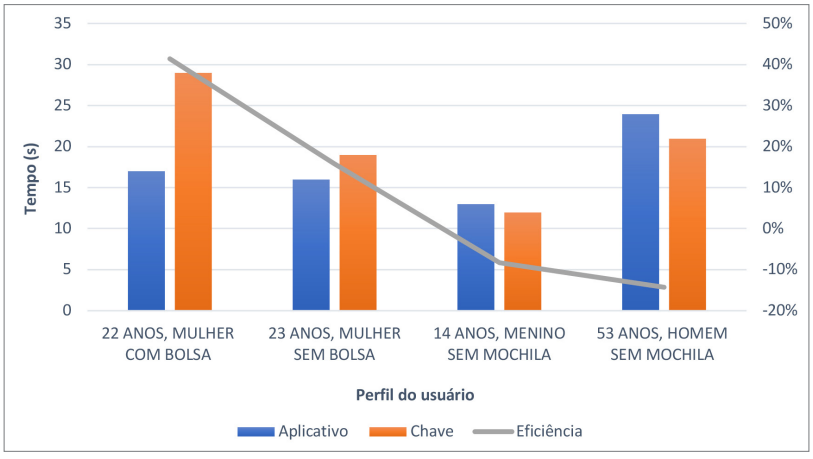

Figura 15 Gráfico de eficiência entre os tempos médios consumidos através da chave e do aplicativo.

Fonte: Elaborada pelos autores.

Embora existam variações de acordo com o perfil do usuário, o tempo médio para acesso pelo aplicativo foi de 17,5 segundos, enquanto, utilizando a chave, obteve-se 20,3 segundos. Nessa situação, pode-se concluir que o aplicativo ofereceu um ganho de tempo de aproximadamente 15% comparado ao procedimento via chave. Tais valores, no entanto, podem variar significativamente dependendo de hábitos de armazenamento de chaves e intempéries; entretanto, a eficiência do aplicativo aumentaria com o seu próprio uso. Mesmo que o desempenho seja pior para certos agentes, esse fator não diminui o mérito do protótipo em relação à segurança e à comodidade oferecida, dado que o acionamento por meio do *app* só será feito por pessoas autorizadas e a partir de qualquer distância, otimizando o tempo de espera em frente à residência ou ao condomínio.

3 RESULTADOS E DISCUSSÃO

Após todos os testes executados e a integração do *app* com a fechadura eletrônica, os resultados obtidos foram:

a) funcionamento devido do aplicativo, com ajustes de conexão em comparação ao planejamento feito na revisão de literatura, mas com a funcionalidade do uso da tecnologia 4G e *Wi-Fi*;
b) tempo médio de acesso do aplicativo com a fechadura eletrônica inferior a dois segundos;
c) possibilidade de aumento da segurança sanitária para acesso aos edifícios que utilizam impressão digital de uso comum em suas entradas;
d) comodidade para abertura do portão onde estiver, desde que haja conexão com a internet.

O aplicativo apresenta uma *performance* adequada, com interface simplificada e intuitiva, o que torna a utilização prática e fácil ao usuário.

Após a instalação do protótipo, uma pesquisa voltada à intenção de uso da solução foi desenvolvida para que os dados relevantes sobre o projeto pudessem ser analisados e para obter a percepção do usuário, contemplando intenção de uso, vantagens, desvantagens observadas na solução, bem como a sua faixa etária. A pesquisa contou com 115 respostas obtidas de 23 usuários; as estatísticas coletadas foram discutidas na sequência.

Na Figura 16 é possível analisar a porcentagem de pessoas que utilizariam a solução, que é equivalente a 80% dos entrevistados.

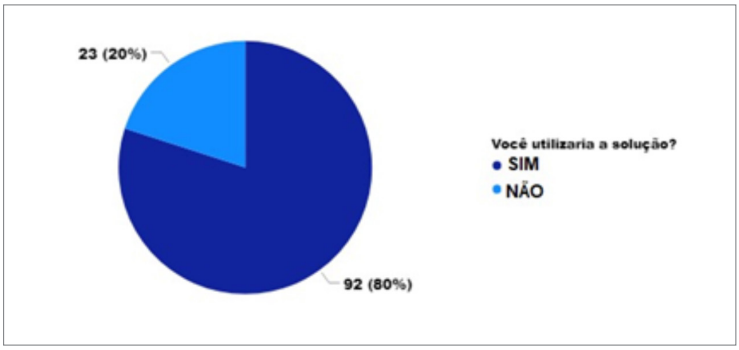

Figura 16 Gráfico comparativo de intenção de utilização da solução.

Fonte: Elaborada pelos autores.

Já na Figura 17, mais de 64% das respostas obtidas apontam que as maiores vantagens da solução são segurança e agilidade.

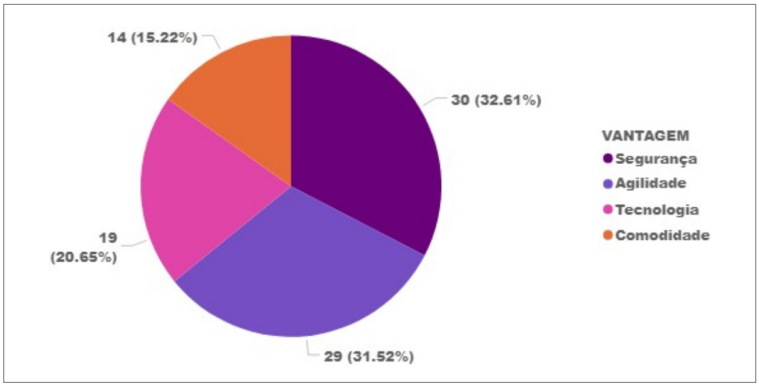

Figura 17 Gráfico demonstrativo de vantagens da solução.

Fonte: Elaborada pelos autores.

Considerando os 20% dos entrevistados que não utilizariam a solução, pode-se observar as maiores desvantagens da solução elencadas na Figura 18.

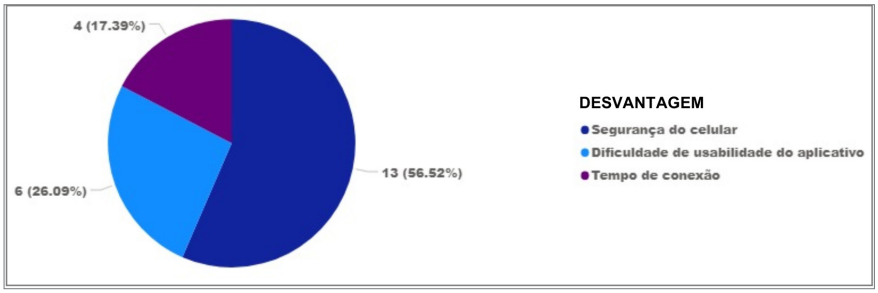

Figura 18 Gráfico de desvantagens X entrevistados que não utilizariam.

Fonte: Elaborada pelos autores.

Outra análise possível refere-se ao número de usuários que não utilizariam o aplicativo, indicados nas faixas etárias entre 15 e 35 anos (Figura 19).

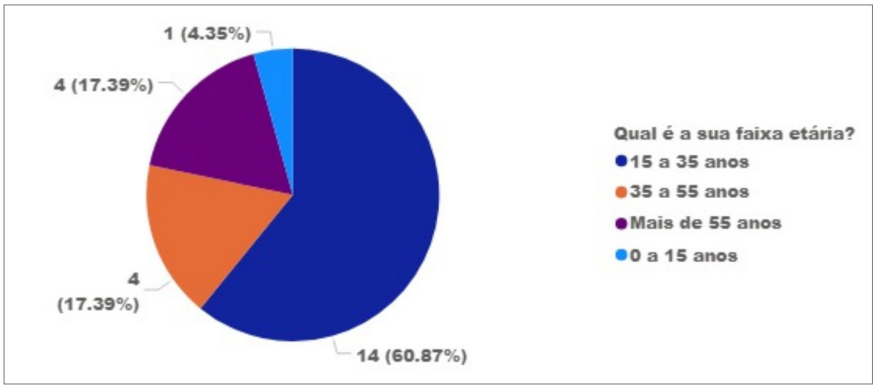

Figura 19 Gráfico de faixa etária de entrevistados que não utilizariam.

Fonte: Elaborada pelos autores.

4 CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo a implementação de uma solução tecnológica focada em proporcionar a facilidade de acesso a edifícios e residências. A ferramenta desenvolvida para a aplicação física dessa solução tecnológica foi a elaboração de um

aplicativo *mobile* desenvolvido através do *framework Flutter*, associado à linguagem de programação *Dart*, conectado ao banco de dados *Firebase* para armazenamento dos dados de acesso. Com essa comunicação, a integração entre o aplicativo e o banco de dados foi possibilitada por meio do microcontrolador NodeMCU e testada, incialmente, em relés, para fins de simulação e, finalmente, implementada em uma fechadura eletrônica de uma residência real.

Os resultados obtidos permitem concluir que o protótipo atendeu aos requisitos estabelecidos de desempenho, com redução média de 15% no tempo de acesso comparado à abertura via chave e tempo de acionamento da fechadura inferior a 2 segundos. Adicionalmente, no universo pesquisado, há clara intenção de uso do aplicativo, com percentual igual a 80%, sendo as principais vantagens percebidas a segurança e agilidade.

DEVELOPMENT OF A MOBILE APPLICATION FOR ACCESS TO BUILDINGS VIA INDIVIDUAL PASSWORDS, FACIAL, AND BIOMETRICS RECOGNITION

ABSTRACT

This article aims to develop a technological solution to allow the entrance of authorized people in residences and condominiums, avoiding a sensitive security flaw during the night and early morning, since some people return from their activities in this period and face a delay that leaves them subject to theft and robbery. There is also the sanitary issue experienced nowadays due to the pandemic generated by Covid-19, in which common areas and surfaces have become the target of great concern regarding the virus's contagiousness. The solution presented was creating an app to enter homes or condominiums through biometric reading from the cell phone itself. This app is developed through the Flutter framework together with an electric lock and a microcontroller, NodeMCU, providing entry authentication for each resident. The results showed that the developed system had a satisfying performance with the connection time presented between the device and the lock being less than two seconds, besides promoting an increase in hygiene at home's entrances, since using an individual app via smartphone.

Keywords: Residences. App. Biometric security.

REFERÊNCIAS

ANDROID STUDIO. 2020. Disponível em: https://developer.android.com/studio. Acesso em: 7 abr. 2020.

COSTA, L. R.; OBELHEIRO, R. R.; FRAGA, J. S. Introdução à biometria. *In*: SIMPÓSIO BRASILEIRO DE SEGURANÇA DA INFORMAÇÃO E DE SISTEMAS COMPUTACIONAIS (SBSeg2006), 6., Porto Alegre, 2006. *Anais* [...]. Florianópolis: Universidade Federal de Santa Catarina, 2006. p. 103-151. Disponível em: http://www.advancedsourcecode.com/minicurso_biometria.pdf. Acesso em: 11 ago. 2021.

FIREBASE. Disponível em: https://firebase.google.com/?hl=pt-br&gclid=Cj0KCQiAqo3-BRDoARIsAE5vnaJdQ_Jl3bOmzQbD1dd7MRRAwFcnwRIRYZWBdZI2P-W8Y_cF8NlJR--8aAjRwEALw_wcB. Acesso em: 28 abr. 2020.

FLUTTER. Disponível em: https://flutter.dev/. Acesso em: 7 abr. 2020.

HU, S. *et al*. Outsourced biometric identification with privacy. *IEEE Transactions on information forensics and security*, v. 13, n. 10, Oct. 2018. Disponível em: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8323394. Acesso em: 11 ago. 2021.

JAIN, A. K.; ROSS, A. A.; NANDAKUMAR, K. *Introduction to Biometrics*. New York: Springer, 2011.

LIMA, F. F. de. *Avaliação de frameworks para o desenvolvimento de aplicações híbridas*. 2019. Trabalho de Conclusão de Curso (Graduação em Engenharia de Software) – Universidade Federal do Pampa, Alegrete, 2019. Disponível em: https://dspace.unipampa.edu.br/bitstream/riu/4224/1/Fernando%20Fortunato%20de%20Lima%20-%202019.pdf. Acesso em: 7 abr. 2020.

OLIVEIRA, S. de. *Internet das Coisas com ESP8266, Arduino e Raspberry Pi*. São Paulo: Novatec, 2017.

ORACLE BRASIL. *Banco de Dados*. [S.d.]. Disponível em: https://www.oracle.com/br/database/what-is-database.html. Acesso em: 28 abr. 2020.

OTTI, C. The past, present and future of biometrics. *Annals of the Faculty of Engineering Hunedoara*, Budapeste, v. 15, n. 2, p. 163-168, May 2017. Disponível em: http://eds.b.ebscohost.com/eds/pdfviewer/pdfviewer?vid=1&sid=bf5f6389-dd5a-461e-98e0--dd90b44b0f17%40pdc-v-sessmgr03. Acesso em: 11 ago. 2021.

RABINEZHADSADATMAHALEH, N.; KHATIBI, T. A novel noise-robust stacked ensemble of deep and conventional machine learning classifiers (NRSE-DCML) for human biometric identification from electrocardiogram signals. *Informatics in Medicine Unlocked*, v. 21, Nov. 2020. Disponível em: https://reader.elsevier.com/reader/sd/pii/S2352914820306195?token=6E6616E5ACD42304255BFF4BB134890D82B34C771F26B33C56DEB1AB53006CAE064A549DA492808FB07CF6233263B28B&originRegion=us-east--1&originCreation=20210811184849. Acesso em: 11 ago. 2021.

REIS, A. C. S. dos. *Um estudo entre modelos de desenvolvimento de aplicações móveis*. 2019. Trabalho de Conclusão de Curso (Graduação em Engenharia de Software) – Universidade Federal do Ceará, Quixadá, 2019. Disponível em: http://repositorio.ufc.br/bitstream/riufc/49707/1/2019_tcc_acsdosreis.pdf. Acesso em: 7 abr. 2020.

VALENTE, J. Brasil é o 3º país em que pessoas passam mais tempo em aplicativos. *Agência Brasil*, Brasília, DF, 16 jan. 2020. Disponível em: https://agenciabrasil.ebc.com.br/geral/noticia/2020-01/brasil-e-o-3o-pais-em-que-pessoas-passam-mais-tempo-em-aplicativos. Acesso em: 6 jun. 2020.

WANG, Y. *et al*. An Automatic Physical Access Control System Based on Hand Vein Biometric Identification. *IEEE Transactions on Consumer Electronics*, v. 61, n. 3, p. 320-327, Aug. 2015. Disponível em: https://ieeexplore.ieee.org/abstract/document/7298091. Acesso em: 11 ago. 2021.

ZHU, L. *et al*. An Efficient and Privacy-Preserving Biometric Identification Scheme in Cloud Computing. *IEEE Access*, v. 6, p. 19025-19033, Mar. 2018. Disponível em: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8325278. Acesso em: 11 ago. 2021.

Contato

Sérgio Luis Rabelo de Almeida
sergioluis.almeida@mackenzie.br

Tramitação

Recebido em fevereiro de 2021.
Aprovado em junho de 2021.