XXI Escola Regional de Redes de Computadores (ERRC). DOI: 10.5753/errc.2024.4566

Artigo Curto/Short Paper

Disponibilizando autenticação via *smartphones* para um sistema de controle de acesso.

Providing authentication via smartphones for an access control system.

Raphael Pahim Noal • raphael.noal@acad.ufsm.br
Universidade Federal de Santa Maria (UFSM)
Tiago Antonio Rizzetti • tiago.rizzetti@ufsm.br
Universidade Federal de Santa Maria (UFSM)

resumo. Este artigo apresenta o desenvolvimento de um aplicativo móvel integrado a um sistema de controle de acesso existente, utilizando *Bluetooth Low Energy (BLE)* e *Flutter*. O objetivo é modernizar o sistema, melhorando a eficiência da autenticação e a experiência do usuário com autenticação em duas etapas e atualizações dinâmicas de permissões.

abstract. This article presents the development of a mobile application integrated with an existing access control system using *Bluetooth Low Energy (BLE)* and *Flutter*. The goal is to modernize the system, improving authentication efficiency and user experience with two-step authentication and dynamic permission updates.

palavras-chave: *Bluetooth Low Energy* • Dispositivo Móvel • *Flutter* • Senha de uso único baseada em tempo (TOTP) • *ESP32*

keywords: *Bluetooth Low Energy* • *Mobile* • *Flutter* • *Time-based one-time password (TOTP)* • *ESP32*

1 Introdução

Este trabalho propõe o desenvolvimento de um aplicativo móvel utilizando o framework *Flutter*, com o objetivo de aprimorar o sistema de controle de acesso físico no Colégio Técnico Industrial de Santa Maria (*CTISM*), integrando-se ao projeto já existente. O aplicativo substituirá a solução anterior, baseada em *Raspberry Pi*, por uma nova implementação utilizando a placa de desenvolvimento *ESP32*, visando aumentar a escalabilidade e a eficiência do sistema.

A proposta não busca apenas migrar as funções já existentes, mas também implementar novas abordagens de autenticação por meio de tecnologias como *Bluetooth Low Energy (BLE)*, garantindo a verificação segura da identidade dos usuários antes de conceder acesso aos ambientes da instituição. Além disso, o projeto considera a expansão futura do sistema para incluir novas funcionalidades, promovendo uma gestão mais eficiente e em tempo real dos espaços físicos e das identidades dos usuários.

2 Referencial Teórico

O *Bluetooth Low Energy (BLE)*, introduzido na versão 4.0 do *Bluetooth*, é ideal para aplicações com exigências rigorosas de consumo energético, como dispositivos alimentados por bateria, e transferência ocasional de pequenas quantidades de dados, como em sensores [1]. Diferente do *Bluetooth* clássico (*BR/EDR*), que opera em 79 canais na faixa de 2,4 *GHz*, o *BLE* suporta comunicação eficiente ponto a ponto, contribuindo para a eficiência energética em dispositivos *IoT*.

O *Flutter*, um *Software Development Kit (SDK)* móvel de código aberto desenvolvido pelo *Google*, revoluciona o desenvolvimento de aplicativos multiplataforma ao fornecer um conjunto abrangente de objetos de interface, renderização e suporte para animações, gráficos, E/S de arquivos e rede [2]. Essa abordagem radical no desenvolvimento de aplicativos visa simplificar e acelerar a criação de interfaces bonitas e funcionais em diversas plataformas móveis.

O artigo [3] propõe o *ABLE*, um sistema de autenticação de dois fatores baseado em *BLE*, que utiliza a detecção de co-localização entre dispositivos para autenticação. Ele explora características como os identificadores de dispositivos e a intensidade do sinal *BLE* por *Received Signal Strength Indication (RSSI)* para verificar a proximidade entre os dispositivos e determinar a legitimidade do acesso, garantindo maior segurança em ambientes dinâmicos. Por outro lado, o artigo [4] define um protocolo de autenticação para *Internet of Things (IoT)*, também utilizando *BLE*, que combina *Universally Unique Identifiers (UUIDs)* e *Time-based One-Time Passwords (TOTP)*. Esse protocolo garante a integridade e autenticidade dos dados transmitidos, focando na eficiência energética e segurança em dispositivos de baixo

custo, sendo testado em um ambiente com *Raspberry Pi*.

3 Metodologia

Conforme mencionado, este projeto propõe a adaptação e modernização de um sistema já existente, adicionando uma interface móvel ao fluxo de controle de acesso físico no Colégio Técnico Industrial de Santa Maria (*CTISM*). O sistema original é composto por quatro componentes principais, e este trabalho foca especificamente na interação com o componente *ESCHA*. Este componente, apresentado isoladamente na Figura 1, é representado por um microcontrolador *ESP32*, modelo *Doit-devkit v1*, que controla o acesso aos ambientes físicos da instituição.

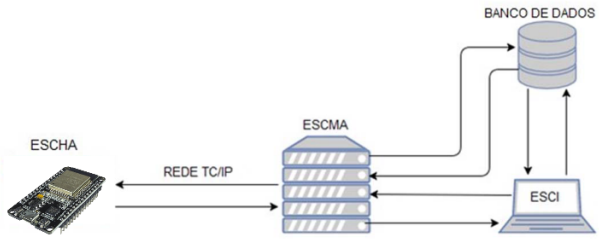

Figura 1. Componentes do sistema ESC.
Fonte: Adaptado de [5].

Ao utilizarmos o *ESP32*, um hardware mais limitado que o *Raspberry Pi*, foi necessário adaptar o código para atender às capacidades do novo dispositivo, impactando o funcionamento do sistema originalmente descrito por [5]. Optamos por incorporar e adaptar o protocolo de autenticação de [4], desenvolvido para o *Raspberry Pi*, ao novo ambiente de baixo custo.

Esse protocolo melhora a segurança ao implementar autenticação em duas etapas, reduzindo o risco de acessos não autorizados. O uso de *UUIDs* exclusivos e senhas temporárias descartáveis (*TOTP*) garante que cada comunicação seja única e segura, mesmo em um ambiente de recursos limitados, como o *ESP32*.

Conforme discutido em [3], o *BLE* possui características que permitem o controle eficiente da distância entre dispositivos. Em [3], o *BLE* é utilizado como parte do sistema de autenticação em dois fatores, no qual a comunicação entre o dispositivo do usuário e o servidor é autenticada por uma combinação de fatores, como credenciais e tokens temporários. Nesse cenário, a ênfase está na verificação da identidade do usuário, dependente da distância entre o dispositivo móvel e o ponto de acesso.

Neste projeto, optou-se por utilizar o *BLE* de forma diferente, aproveitando suas capacidades de medição de distância para definir uma área de operação limitada, em vez de usá-lo apenas para autenticação. Será implementado um serviço sobre o *Generic Attribute Profile (GATT)* do *BLE* no sistema existente. O *GATT* permite que dispositivos *BLE* se comuniquem utilizando serviços identificados por *UUIDs*. O sistema será configurado com duas funções principais: autenticação da comunicação do usuário e controle da abertura da fechadura.

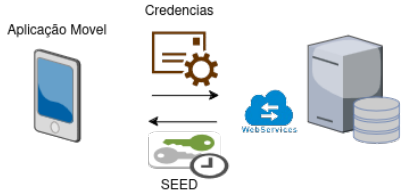

Figura 2. Obtendo SEED.
Fonte: Autor (2024).

A Figura 2 mostra como a nova metodologia de autenticação funciona: o usuário, usando suas credenciais de login, receberá uma *SEED* temporária ao logar no aplicativo como forma de autenticação de segundo fator. Em paralelo, como demonstra a Figura 3, o *ESP32* deverá armazenar em um banco de dados *SQLite* local uma lista com todas as *SEEDs* que possuem permissões de acesso.

Figura 3. Sistema de Auteticação Assincrono PT1.
Fonte: Autor (2024).

Dessa forma, o dispositivo pode realizar a verificação de acesso diretamente, sem a necessidade de consultar o servidor para cada tentativa de autenticação, como apresentado na Figura 4.

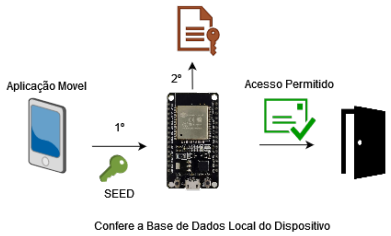

Figura 4. Sistema de Auteticação Assincrono PT2.
Fonte: Autor (2024).

Esse método permite que o sistema *ESCHA* funcione de forma mais autossuficiente, reduzindo a dependência

da conexão de rede. Apenas em casos onde o *ESP32* não consiga identificar a *SEED* do usuário localmente, ele enviará uma requisição ao servidor.

4 Desenvolvimento

Embora ainda se encontre em um estágio inicial, o sistema da aplicação já implementa a maioria de suas funcionalidades básicas. Isso inclui a adaptação do banco de dados do sistema existente para suportar a criação e vinculação da *SEED* ao usuário durante o processo de autenticação por login, como mostrado na Figura 5 na tela da esquerda, assim como a reformulação completa dos *web services* utilizados no sistema antigo baseado em *Raspberry Pi*, representados nas Figuras 2 e 3.

O sistema realiza validações periódicas do tempo de vida das *SEEDs*, e, ao efetuar o login, o aplicativo renova automaticamente a *SEED*. Adicionalmente, o usuário possui a opção de atualizá-la manualmente através de um botão na interface principal, como mostrado na Figura 5 na tela da direita.

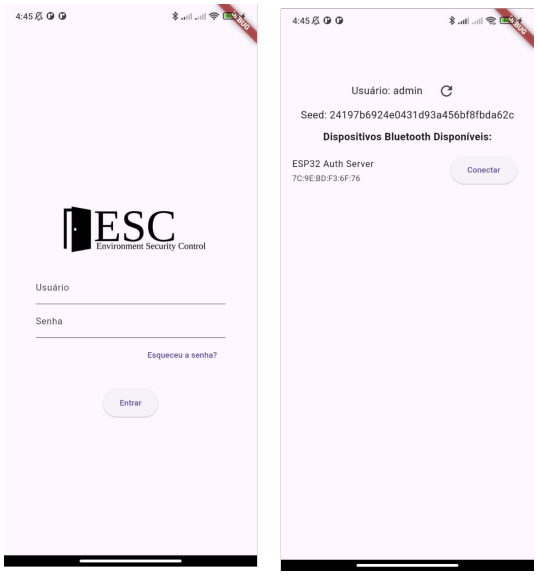

Figura 5. Prototipo Aplicativo.
Fonte: Autor (2024).

Na parte referente ao Servidor *GATT*, foi desenvolvida uma versão preliminar do serviço em um dispositivo *ESP32* autônomo, sem ligação com o sistema atual, utilizando o *FreeRTOS*, um sistema operacional de tempo real leve e eficiente que permite ao *ESP32* executar várias tarefas ao mesmo tempo. Esse sistema operacional, muito utilizado em sistemas embarcados, é essencial para o gerenciamento de múltiplas operações simultâneas, como a comunicação *BLE* e *Wi-Fi*, e é responsável pelo controle dos recursos do sistema. Esse tipo de abordagem já é aplicada no sistema atual, o que torna eficiente a forma como o *ESCHA* implementa todas suas funcionalidades.

Na interface principal da aplicação em *Flutter*, foi definido um tempo de 4 segundos, durante o qual o aplicativo irá escanear em busca de dispositivos *BLE* com a mesma *UUID* definida no serviço do servidor *GATT*, como vemos o *ESP32* listado na Figura 6.

Figura 6. Lista de Dispositivos Encontrados.
Fonte: Autor (2024).

No caso, a versão do serviço apresentado no *ESP32* nos testes até agora não apresentou nenhuma funcionalidade dos serviços propostos por ter se focado mais em adaptar o sistema presente para o suporte ao *TOTP* e o desenvolvimento de um protótipo funcional básico da aplicação *mobile*.

5 Considerações Finais e Trabalhos Futuros

Este trabalho moderniza o sistema de controle de acesso existente, integrando o *ESP32* e o *Flutter* para maior eficiência e escalabilidade. A substituição pelo *ESP32* reduziu custos sem perder funcionalidade, enquanto o *Flutter* permitiu criar rapidamente um aplicativo móvel multiplataforma com interface intuitiva, melhorando a experiência do usuário na autenticação.

O principal ganho é a melhoria na segurança e eficiência, assim como o uso do *BLE* para delimitação geográfica e sincronização local de credenciais, permitindo operação sem conexão contínua. Diferentemente de trabalhos anteriores, esta solução adapta tecnologias para hardware de baixo custo como o *ESP32*. Ainda são necessárias modificações no código do *ESCHA* para incorporar novas funcionalidades e garantir a sincronização eficiente das *SEEDs*. Recomenda-se também a reformulação da interface da aplicação para melhorar a usabilidade.

Como trabalhos futuros, planeja-se implementar o funcionamento do aplicativo em segundo plano, permitindo que smartphones atuem como cartões de aproximação sem a necessidade de abrir a aplicação. Além disso, desenvolver mecanismos de administração via *BLE* para obter dados internos do *ESP32*, melhorando a prevenção e correção de falhas, assegurando a evolução e robustez do sistema.

Referências

1 Afaneh, M. *Intro to Bluetooth low energy*. Novel Bits, 2018.

2 Windmill, E. *Flutter in action*. Simon e Schuster, 2020.

3 He, Y. *et al.* ABLE: Zero-effort two-factor authentication exploiting BLE co-location. *In*: IEEE. 2022 IEEE Wireless Communications and Networking Conference (WCNC). 2022. P. 992–997.

4 Eichner, A.; Silva, N. da; Rizzetti, T. A. Definindo um protocolo de autenticação utilizando bluetooth low energy para dispositivos no conceito de iot. *In*: SBC. ANAIS da XVII Escola Regional de Redes de Computadores. 2019. P. 97–104.

5 Pedrozo, W. F. *et al.* Uma arquitetura para Sensoriamento e Tratamento de Eventos voltada à Área de Segurança para Controle e Rastreio de Usuários em Ambientes Físicos. *In*: UFSM. ANAIS da 15ª ERRC. Santa Maria, RS, Brasil, 2017. P. 22–30. Disponível em: https://www.ufsm.br/app/uploads/sites/360/2019/06/ANAIS_ERRC_2017.pdf.