Felipe Augusto Pereira dos Santos

Desenvolvimento de um gerenciador de senhas via *hardware*

Bauru

2023

Felipe Augusto Pereira dos Santos

Desenvolvimento de um gerenciador de senhas via *hardware*

Trabalho de Conclusão de Curso apresentado como parte dos requisitos para obtenção do título de bacharel em Engenharia Elétrica junto ao Centro Universitário Sagrado Coração.

Aluno: Felipe Augusto Pereira dos Santos
Orientador: Prof. Dr. Tiago Forti da Silva

Bauru

2023

Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD

Santos, Felipe Augusto Pereira Dos
S237d

Desenvolvimento de um gerenciador de senhas via hardware / Felipe Augusto Pereira Dos Santos. -- 2023.
23f. : il.

Orientador: Prof. Dr. Tiago Forti da Silva

Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica) - Centro Universitário Sagrado Coração - UNISAGRADO - Bauru - SP

1. Gerenciador de Senhas. 2. Hardware Embarcado. 3. Dispositivo Offline. I. Silva, Tiago Forti da. II. Título.

Elaborado por Lidyane Silva Lima - CRB-8/9602

FELIPE AUGUSTO PEREIRA DOS SANTOS

DESENVOLVIMENTO DE UM GERENCIADOR DE SENHAS VIA *HARDWARE*

Trabalho de Conclusão de Curso apresentado como parte dos requisitos para obtenção do título de bacharel em Engenharia Elétrica junto ao Centro Universitário Sagrado Coração.

Aprovado em: 01/12/2023

Banca examinadora:

Prof. Dr. Tiago Forti da Silva (Orientador)

Centro Universitário Sagrado Coração

Prof. Dr. Danilo Sinkiti Gastaldello

Centro Universitário Sagrado Coração

Resumo

A crescente conectividade na vida moderna tornou o uso de senhas fortes e não repetitivas um problema crítico de segurança. No entanto, gerenciar várias senhas e alterá-las regularmente é um desafio, resultando em baixa conformidade do usuário e no desvio de políticas de segurança por meio do uso de senhas semelhantes ou notas escritas facilmente acessíveis. Uma possível solução para esse problema é a utilização de sistemas centralizados de gerenciamento de senhas, como soluções híbridas ou baseadas em nuvem, que melhoram a qualidade e a diversidade das senhas. No entanto, essas soluções também introduzem novas vulnerabilidades de segurança, incluindo ataques de *phishing* e violações de servidores. Para melhorar a segurança e manter a comodidade do dia a dia, foi realizado neste estudo o desenvolvimento de um dispositivo *off-line* para gerenciamento seguro de senhas, permitindo o registro, modificação, exclusão e entrada segura de senhas em computadores e smartphones. Seu desenvolvimento inclui a configuração inicial, armazenamento de senhas no ESP32, tela de configuração das senhas e do dispositivo, emulação de teclado com Arduino Pro Micro, integração entre ESP32 e Arduino Pro Micro, testes e validação, em suma, mesmo como um MVP (Produto Mínimo Viável), atende aos objetivos iniciais, oferecendo facilidade de uso para senhas e uma camada adicional de segurança.

Palavras-chave: Gerenciador de Senhas. Hardware Embarcado. Dispositivo Offline.

Abstract

The increasing connectivity in modern life has turned the use of strong and non-repetitive passwords into a critical security issue. However, managing multiple passwords and changing them regularly poses a challenge, resulting in low user compliance and deviation from security policies through the use of similar passwords or easily accessible written notes. A possible solution to this problem is the use of centralized password management systems, such as hybrid or cloud-based solutions, which enhance the quality and diversity of passwords. Nevertheless, these solutions also introduce new security vulnerabilities, including phishing attacks and server breaches. To enhance security and maintain everyday convenience, this study undertook the development of an offline device for secure password management, allowing the registration, modification, deletion, and secure entry of passwords on computers and smartphones. Its development includes the initial setup, password storage on ESP32, password and device configuration screen, keyboard emulation with Arduino Pro Micro, integration between ESP32 and Arduino Pro Micro, testing, and validation. In summary, even as a Minimum Viable Product (MVP), it meets the initial objectives, providing ease of use for passwords and an additional layer of security.

Keywords: Password Manager. Embedded Devices. Offline Device.

Lista de ilustrações

Sumário

1 Introdução

Conforme Mynatt et al. (2010) a alta conectividade da vida moderna tornou o uso de senhas fortes e não repetidas uma importante questão de segurança. Entretanto, a dificuldade de gerenciar uma quantidade crescente de senhas e alterá-las periodicamente resulta em uma baixa adesão dos usuários, implicando em formas de *bypass* (em tradução direta seria o equivalente a desviar ou contornar, em uma tradução mais eficiente seria o ato de evitar uma medida de segurança ou de controle) dessas políticas. Como por exemplo, o uso de uma mínima alteração entre uma senha e outra ou anotar a senha em um caderno de fácil acesso.

Uma possível solução para esse problema é a utilização de um sistema centralizado de gerenciamento de senhas, dentro deste contexto, apareceram as soluções de gerenciamento de senhas em nuvem ou híbridas (salvas em um dispositivo móvel e sincronizado com a nuvem). Isso resulta em uma melhoria da qualidade das senhas, já que permite o aumento de sua complexidade e maior diversificação. Ao utilizar de soluções conectadas a internet, também é adicionado um novo contexto para os pontos de falha de segurança, como por exemplo os ataques do tipo *phishing* (uma forma de ataque virtual, onde o ataque emula a tela de login ou outra tela do sistema com o objetivo de capturar dados do usuário) aos serviços de controle de senhas ou até mesmo a invasão de seus servidores.

Ainda dentro desse contexto, segundo Li et al. (2014) existem dentre os pontos a serem reforçados pelos gerenciadores de senha, as seguintes fraquezas:

- Segurança do acesso principal - Definir uma boa política para a criação das credenciais para acessar o gerenciador de senhas e garantir um acesso seguro.
- Segurança da base de dados do gerenciador de senhas - Proteger contra vazamentos dos dados e armazenar os dados de forma segura (encriptada).
- Integridade do gerenciador e do site do cliente - Garantir que o site, extensão ou aplicativo acessado seja o oficial e não de um atacante, bem como do site que o cliente deseja acessar.
- Não correlacionar o usuário aos identificadores dos sites - Não rastrear o que o cliente está acessando ou realizado, apenas guardar os dados e interagir com os sites o suficiente para permitir a identificação de qual credencial utilizar e inseri-la no site.

Para melhorar essa segurança e manter a facilidade no dia a dia, este trabalho propõe a criação de um dispositivo *offline* capaz de fazer o gerenciamento dos cadastros das senhas (criar, alterar e deletar), bem como inserir a senha nos computadores ou nos *smartphones*. Para ser possível realizar a seleção da senha, é proposto o uso de uma tela lcd e um *encoder* com botão para poder selecionar a senha e mandar executar, para o cadastro, planeja-se utilizar um ponto de conexão *Wi-Fi* gerado pelo microcontrolador em conjunto com uma página web. Por fim, para os microcontroladores, será utilizado um ESP32 para o ponto *Wi-Fi* e gerenciamento das senhas, e um *Arduino* Pro Micro para a emulação do teclado e inserção da senha no dispositivo escolhido.

2 Objetivos

2.1 Objetivos Gerais

Gerar um MVP (*Minimum Viable Product* ou em português Minimo Produto Viável) de um gerenciador de senhas via *hardware*.

2.2 Objetivos Específicos

a) Permitir o cadastro de senhas em um dispositivo sem conexão com a rede.

b) Emular um teclado com o objetivo de inserir a senha selecionada no computador ou celular.

c) Ter uma autenticação para permitir o cadastro da senha e posteriormente a sua inserção.

3 Revisão Bibliográfica

Nesta revisão, abordaremos os desafios atuais em segurança de senhas, explorando soluções centralizadas na nuvem e os riscos associados. Além disso, examinaremos dispositivos *offline* como alternativas inovadoras para gerenciamento seguro de senhas, conectando-se aos objetivos de identificar lacunas no conhecimento e promover avanços nesse campo.

3.1 Gerenciador de Senhas

Conforme Oesch e Ruoti (2020), o gerenciador de senhas tem o objetivo de tornar o acesso a diversos sites em algo mais conveniente para os usuários, focando em ter apenas uma credencial (conjunto de usuário, senha e plataforma para autenticar) e a partir dela se autenticar nos outros sites, sendo cada um com um usuário e senha diferentes, todos armazenados pelo gerenciador de senhas.

Segundo Stobert e Biddle (2014) gerenciadores de senhas são atualmente as melhores ferramentas tecnológicas para ajudar o usuário final a manter suas autenticações digitais seguras, tendo em vista que os gerenciadores podem tanto armazenar as credenciais em forma de carteira (arquivo encriptado com todos os dados) ou de forma *online*. Eles também são genéricos o bastante para funcionar na maioria dos sites, logo, não é necessário ter que esperar o site se adequar às melhores práticas de segurança ou trocar seu sistema para o SSO (*Single Sign-On* ou Acesso unitário) que seria forçar uma nova senha para cada acesso, seja por *e-mail* ou por aplicativo.

Eles também atacam o problema da quantidade de senhas onde é necessário utilizar diariamente, sendo que apenas tem-se de decorar uma senha forte, que seria a autenticação do gerenciador de senhas, as outras serão criadas e armazenadas por ele. Apesar de ser uma boa característica, isso também pode ser um ponto de entrada para o atacante, já que tendo acesso a apenas uma senha pode-se controlar o acesso a vários sites e ferramentas.

De acordo com View, Summers e Bosworth (2004) , uma outra técnica utilizada pelos usuários quando não tem um gerenciador de senhas é justamente anotá-las, quanto ao ato de anotar as senhas em um meio físico e guardá-las em um local seguro, como por exemplo em casa, não temos uma real falha de segurança, pois a chance de alguém sem permissão ter acesso a essas anotações é muito baixa, porém há algumas deturpações quanto a essa prática, por exemplo anotar esses dados em um arquivo digital e deixá-lo *online* ou ainda deixar esse caderno em um local de fácil acesso como na mesa do trabalho, nesse caso, existe uma falha grande de segurança, já que pessoas não autorizadas podem sim ter fácil acesso à esses dados.

Entende-se com isso que o gerenciador de senhas supre uma grande necessidade dos usuários finais, porém deve-se sempre balancear seu uso, levando em conta os vazamentos e as invasões já realizadas neles como um aprendizado de que uma proteção de dois ou mais fatores se faz necessária nos dias atuais.

3.2 NFC

A tecnologia NFC (*Near Field Communication*) conforme Mohammadinodoushan et al. (2021) foi um desenvolvimento em conjunto da Philips e Sony no final de 2002 para comunicação sem contato, ele é um protocolo de curto alcance que funciona com base na comunicação *Half Duplex*, que provê uma comunicação segura e fácil entre diversos dispositivos.

Ele é diferente do RFID (*Radio Frequency Identification*, ou Identificação por Rádio Frequência) pois no caso do NFC há um acoplamento indutivo entre os dispositivos permitindo o uso em curta distância apenas (alguns centímetros), já no caso do RFID há apenas e transmissão de Rádio Frequência com suas modulações permitindo um alcance muito maior. Existem dois tipos de comunicação NFC sendo uma o modo ativo, onde os dois dispositivos geram um campo RF (Rádio Frequência), e o modo passivo, onde um dos dispositivos gera a RF e o outro não.

Conforme discutido por Coskun, Ozdenizci e Ok (2013), entre as aplicações do NFC no cotidiano existem as com foco na área da saúde, principalmente em dispositivos de *log* de dados e monitoramento, tendo em vista seu baixo consumo e facilidade de interação pelo usuário final; aplicações em ambientes inteligentes com o foco em facilitar a comunicação do mundo físico com o mundo virtual, como por exemplo o controle de dispositivos inteligentes e acionamento de rotinas do usuário; aplicações de compartilhamento de informações e troca de dados, existem atualmente desde o compartilhamento de dados de contato até o envio de pequenos arquivos ou *links*.

Também é utilizado em meios de pagamento e *tickets*, com o mercado tendo que se adaptar as novidades tecnológicas, de acordo com Allyson, Lakshmi e Packialatha (2015) existe a adoção do NFC em cartões de crédito e débito, armazenar cupons de desconto das lojas, bem como em sistemas de controle de catracas, por exemplo para passagens de ônibus locais e garagens.

4 Metodologia

Neste capítulo é apresentada a metodologia utilizada para criar um gerenciador de senhas *offline*, onde o protótipo utiliza um ESP32 para armazenar e gerenciar (listar, criar, editar e excluir) as senhas, uma tela LCD com Encoder para selecionar e dar o comando de inserir a senha no computador, um Arduino Pro Micro para emular um teclado e inserir a senha no computador e um módulo RFID para facilitar a autenticação para uso do dispositivo. Descreve-se a seguir alguns detalhes dos componentes utilizados e funções necessárias para o uso desse dispositivo.

4.1 ESP32

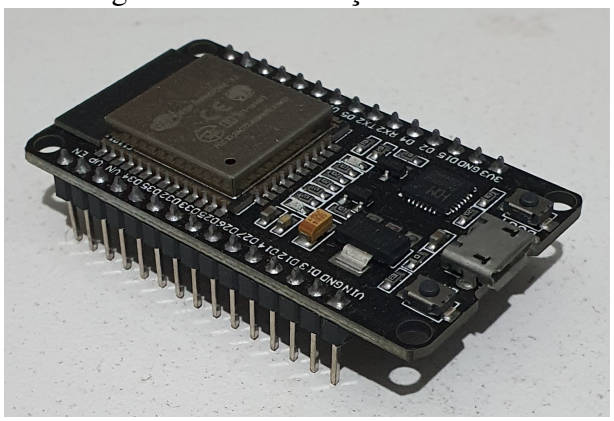
Conforme a documentação da Expressif (2022), o ESP32 é um SoC (*System on Chip*, ou Sistema em chip), também chamado de microcontrolador, ele possui comunicações externas mais abrangentes, como o *Wi-Fi* e *Bluetooth* nativos. Também possui dois núcleos de processamento, 36 pinos de GPIO (*General Pourpose Input and Output*, ou em português, Entradas e Saídas de Uso Geral), e conta com uma vasta biblioteca padrão, para os mais diversos fins, desde uso do *Wi-Fi* até o uso de sensores específicos e protocolos de comunicação.

De acordo com Babiuch, Foltýnek e Smutný (2019), o ESP32 foi desenvolvido pela *Expressif Systems* que atualmente produz algumas variações desse chip, cada qual com a sua peculiaridade, com desde integração com cartões SD até acoplado à câmeras de vídeo. Seus núcleos podem ser controlados separadamente e suportam o uso do RTOS (*Real Time Operating System* ou Sistema Operacional em Tempo Real) que seria a capacidade de controlar individualmente cada função com base no tempo de uso dos processadores, permitindo a criação de componentes mais robustos a falhas e mais eficientes quanto ao gerenciamento de tempo.

Segundo Eswar (2021), para a sua programação, existem em resumo duas opções semelhantes e uma em MicroPython, o uso do Arduino IDE, um ambiente *open-source* e de fácil prototipagem para os microcontroladores da Atmel, sua linguagem de programação é bem semelhante ao C++, porém com algumas modificações para facilitar a interação entre os *hardwares*. Há também a opção de utilizar o ESP-IDF da própria Expressif, ele tem um foco mais profissional e por isso utiliza o C++ em vez do *Wiring*, permitindo uma complexidade maior na aplicação, bem como maior controle sobre os processos e tarefas rodando entre os dois núcleos do ESP32. Sobre o uso do MicroPython, é uma versão modificada do Python para rodar em microcontroladores, é pensado em ser mais leve e tem foco em reduzir a curva de aprendizado para a programação.

Neste projeto, ele foi utilizado para armazenar os dados de autenticação (Descrição, Usuário e senha), criar uma interface WEB para configuração desses dados e demais características do sistema, bem como controlar a tela LCD e *encoder* para seleção que serão descritos mais adiante.

Figure 1: Demonstração do ESP32

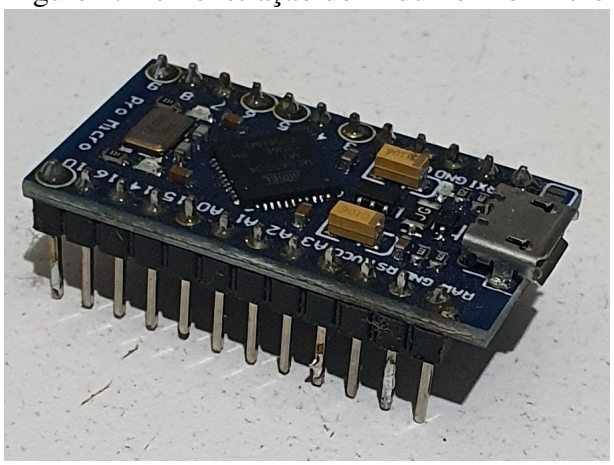


Fonte: Elaborado pelo autor.

4.2 Arduino Pro Micro

O Arduino Pro Micro, segundo Arduino (2023) é um microcontrolador projetado pela empresa Arduino (com o Hardware aberto, ou seja, todo o projeto elétrico é disponibilizado ao publico de forma gratuita, podendo conter restrições de uso), esse *hardware* conta com recursos especiais para emulação de teclado e mouse, permitindo a inserção das senhas no computador. Neste projeto, ele tem a função de receber os comandos do ESP32 via interface serial e emular a entrada de caracteres de um teclado, enviando assim a senha selecionada no produto para o computador.

Figure 2: Demonstração do Arduino Pro Micro



Fonte: Elaborado pelo autor.

4.3 Tela LCD e *Encoder*

A tela LCD conforme explicado por WatElectronics (2021) é uma sigla para *Liquid Crystal Display*, que em português significa Tela de Cristal Líquido, no caso do dispositivo utilizado neste projeto, é uma tela de 16x2, que nos indica que é uma matriz com 16 caracteres por linha (coluna) e com duas linhas, essa tela tem por função neste projeto a listagem das senhas (pré-configuradas na página WEB gerada pelo ESP32) e a seleção da senha.

Há também o Encoder Ky-040, que tem como função converter um movimento rotativo em onda quadrada, ele tem uma resolução de 20 pulsos por revolução, neste modelo existe também um botão ao pressionar o encoder para baixo. Esse módulo de encoder tem por objetivo receber a entrada de comandos do usuário (girar para mudar o item selecionado e pressionar para confirmar) em sinais para o ESP32 computar qual senha foi selecionada.

Figure 3: Display LCD com *Encoder* exibindo as descrições das senhas

Fonte: Elaborado pelo autor.

4.4 Encriptação da memória e *efuse*

Uma função que foi citada como complementar, porém necessária para um produto final, é a de encriptação dos dados e do programa armazenados na memória do ESP32 para a proteção do usuário, conforme página de documentação da Expressif (2023), existem quatro espaços na memória que podem apenas passar do bit 0 para o bit 1 e não podem voltar, essas memórias são de uso único para proteger informações extremamente importantes e necessárias para projetos profissionais utilizando o ESP32.

Este projeto é complementado por duas dessas memórias, a *EFUSE_BLK1* que é responsável por armazenar a chave de encriptação da memória flash e a *EFUSE_BLK2* que é utilizada para guardar os dados da chave do *Security Boot*, que é a função responsável por proteger o programa armazenado no microcontrolador.

5 Desenvolvimento do Software

5.1 Configuração Inicial

O primeiro passo executado foi configurar o ambiente de desenvolvimento para trabalhar com o ESP32 e o Arduino Pro Micro. foram listadas e instaladas as bibliotecas necessárias para o desenvolvimento de firmware para cada um desses dispositivos, sendo o Arduino IDE o ambiente de desenvolvimento escolhido para a programação de ambos os controladores.

5.2 Armazenamento de Senhas no ESP32

Utilizando a linguagem de programação Arduino, foi desenvolvido o código para o ESP32, sendo necessária a criação de uma estrutura de dados para armazenar as senhas de forma interna no ESP32, e com foco em posteriormente a criação do MVP (Mínimo Produto Viável) tornar esse armazenamento mais seguro com a criptografia dos dados e autenticação.

Os campos selecionados para armazenamento são os seguintes:

- **Descrição:** Para que o usuário consiga por uma descrição de onde a senha será utilizada.
- **Usuário:** Para que o usuário consiga se lembrar do usuário cadastrado e caso queira utilizar a opção de Tab que será descrita no decorrer dos itens.
- **Senha:** A senha propriamente dita, que será digitada via Arduino no computador ou smartphone do usuário.
- **Tab:** Será uma opção de caso o usuário queira apenas inserir a senha ou inserir o usuário, o comando Tab do teclado, para mudar de campo de entrada, e depois inserir a senha.

Foi utilizado, para esse fim, a biblioteca ArduinoNVS que faz a conversão da biblioteca NVS (*Non Volatile Storage*, ou armazenamento não volátil) para o uso dentro do ambiente do Arduino, já que esse é o sistema de armazenamento padrão do ESP32 via ESP-IDF que seria seu ambiente de desenvolvimento criado pela desenvolvedora do chip.

Foi gerado o CRUD (acrônimo de *Create, Read, Update and Delete*, ou em português, Criar, Ler, Atualizar e Deletar), que seria um modelo de gerenciamento de dados, onde se tem as funções básicas para manipular as informações armazenadas de forma digital.

Bem como foram geradas funções complementares nessa etapa, sendo elas a de encontrar o próximo ID de senha válido (e por consequência o último ID cadastrado), a função de listagem da descrição das senhas cadastradas.

5.3 Tela de configuração das senhas e do dispositivo

Foi pensado que para facilitar o cadastro, edição ou exclusão das senhas, bem como a edição das configurações do dispositivo, seria ideal ter uma tela maior e mais fácil de visualizar. Dentro desse contexto, foi especificado de o ESP32 gerar um ponto de conexão *Wi-Fi* e a partir dele ter as telas para configuração. Com a segurança em mente, também foi pensado em ter um botão físico para habilitar o *Wi-Fi*, reduzindo as chances de permitir uma edição por terceiros.

Dentro do conceito de desenvolvimento com foco em MVP, foram criadas telas em um site de prototipagem, focando em gerar telas de forma rápida e intuitiva, facilitando a geração dos

códigos web (HTML, JavaScript e CSS), foi utilizado o site webflow.com para esse fim. Sobre as telas criadas, seguem abaixo suas descrições.

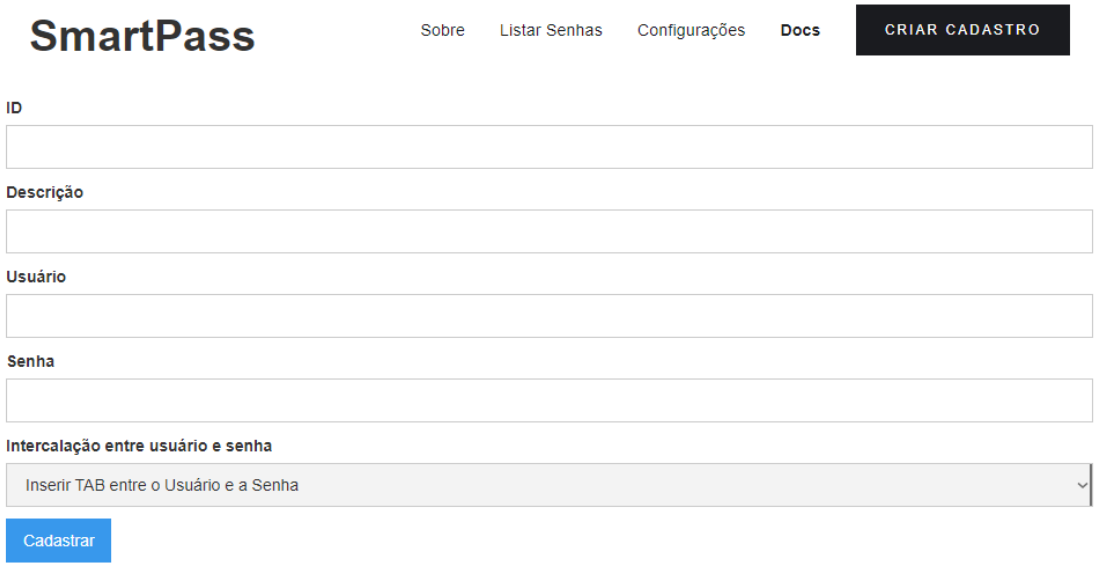
- **Listagem de Senhas:** Utilizado para que o usuário veja as senhas já cadastradas e selecioná-la para caso queira editar ou excluir, segue a seguir uma captura de tela com o layout da página.

Figure 4: Tela de Listagem de Senhas

Fonte: Elaborado pelo autor.

- **Cadastro de senhas:** Essa tela foi criada com o objetivo de servir como tela de cadastro de uma nova senha, mas também como tela de edição de uma senha já existente.

Figure 5: Tela de Cadastro e Edição de Senhas



Fonte: Elaborado pelo autor.

- **Configuração do Dispositivo:** Foi feita uma tela para que o usuário consiga configurar o dispositivo em si, como por exemplo, alterar o nome e senha do *Wi-Fi*, bem como recadastrar a TAG responsável pelo desbloqueio da inserção de senhas, segue abaixo a ilustração da tela criada.

Figure 6: Tela de Configurações

Fonte: Elaborado pelo autor.

- **Sobre:** Por fim, foi criada uma tela para falar do projeto e do autor, a fim de permitir uma forma de contato e de transmitir o motivo da criação do projeto, conforme a captura de tela abaixo.

Figure 7: Tela Sobre

Fonte: Elaborado pelo autor.

5.4 Uso do *encoder* e tela LCD para manipular as senhas cadastradas

Foram utilizados um encoder e uma tela LCD acoplados ao ESP32 para poder listar as senhas e selecionar a que deseja-se enviar para o celular ou notebook, usando o *encoder* para mudar entre senhas e ao pressioná-lo selecionar a senha ao enviar.

Caso seja gerado o produto final, serão necessárias algumas modificações, como por exemplo, informar que é necessário autenticar a cada inicio do ESP para então listar as senhas e posteriormente selecioná-las para envio ao dispositivo conectado.

Figure 8: Display LCD com *Encoder* exibindo as descrições das senhas

Fonte: Elaborado pelo autor.

5.5 Emulação de Teclado com Arduino Pro Micro

Utilizando a plataforma Arduino, foi desenvolvido um código para o Arduino Pro Micro para receber os comandos do ESP32 e transmiti-los ao computador via emulação de teclado, permitindo efetuar o login no site desejado.

5.6 Integração entre ESP32 e Arduino Pro Micro

Para integrar o ESP32 e o Arduino Pro Micro, foi estabelecida uma comunicação serial entre os dois, no Arduino utilizando a biblioteca SoftwareSerial e no ESP32 o EspSoftwareSerial que é uma adaptação para o ESP da biblioteca utilizada no Arduino. Neste modelo, o ESP vai enviar os caracteres para o Arduino e este por sua vez, vai ler caractere por caractere e emular o teclado para envio de dados.

5.7 Testes e Validação

Após a implementação do sistema, foram realizados testes para verificar o seu funcionamento e validar a eficiência do gerenciador de senhas.

Foram testados de forma automática (gerada uma função interna no código) as funções de apagar todas as senhas, criar novas senhas, editar as senhas, excluir algumas das senhas criadas e listar as senhas, todas essas funções possuem uma saída de *LOG* (em português teria a tradução de diário, é uma forma de externar o que o dispositivo está realizando para podermos entender seu comportamento em uma eventual correção de problemas) via saída serial e sem interação com o página WEB de configuração, o foco dessa automação foi confirmar que todas as funções relacionadas a manipulação dos cadastros de senha dentro do programa estão funcionando de

acordo com o esperado.

Foram realizados os testes manuais (testes simulando a interação real de usuários) nas telas de configuração e na utilização do produto via tela LCD e *encoder* foram criados cadastros na tela de cadastro, que também e utilizada para edição e remoção, o que difere a edição da crição é que a URL passada, ao clicar para editar é passado o ID do cadastro dia método GET.

Também foram testadas a inserção de senhas no projeto, onde são listadas as senhas na tela LCD, depois é selecionada a senha a ser inserida via encoder (rotaciona para selecionar a descrição da senha exibida na tela e pressiona o encoder para confirmar a seleção), após esses procedimentos, é feito o envio da senha selecionada via seria RS232 do ESP32 para o Arduino Pró Micro e por fim, o Arduino realiza a emulação de um teclado e envia a senha para o computador ou *Smartphone*, nesses trecho foi essencial checar se a senha chegou corretamente no dispositivo e na ordem correta, bem como testar várias senhas e com IDs (cadastros) disferentes, dessa forma foi garantindo que as operações ocorreram de forma correta.

Dentro desse escopo, é essencial dentro do programa entregar textos e comandos auto-explicativos para o usuário, bem como limitar as entradas de dados do usuário aos valores permitidos pelo programa, por exemplo, todos os campos de texto, com exceção da descrição (já que será utilizada na tela LCD que mostra apenas 16 caracteres), tem o tamanho de 23 bytes, com o foco em garantir que o será possível manter os dados armazenados de forma correta no ESP32.

5.8 Resultados Finais

O produto, por ser um MVP, teve algumas funções não produzidas, tendo em vista a velocidade de desenvolvimento e de testes, sendo estes listados em tópicos abaixo para uma futura melhoria:

- **Módulo NFC para autenticação:** Esse módulo seria utilizado para debloquear" o produto, tendo em vista o uso de cartões ou TAGs NFC para poder autenticar esse desbloqueio sem a necessidade do usuário memorizar a senha.
- **Encriptação dos dados armazenados na memória Flash do ESP32:** Por ser um teste de produto, não teve-se tempo hábil de gerar um ambiente totalmente seguro para o armazenamento das senhas, porém exitem técnicas de encriptação da memória flash do ESP32, gerando uma maior segurança ao cliente.
- **Função de backup das senhas:** Também buscando gerar um produto de testes, não foi criada uma função que realize o backup das senhas, o que ajudaria o cliente a duplicar o dispositivo, ou mesmo salvar os dados dele em algum local seguro para caso o dispositivo falhe.
- **Autenticação ao acessar o site de configuração:** Pode-se gerar uma tela de login ao acessar o site de configuração interna do produto.

Tendo esses dados em mente, o produto está funcionando de acordo com o esperado e gera uma facilidade de uso das senhas, principalmente as raramente utilizadas, bem como gera uma segurança extra por dificultar o acesso as mesmas, já que anteriormente elas eram armazenadas em um bloco de notas físico.

Tem-se que sempre se atentar ao retorno dos testes e buscar melhorias, principalmente ao lidar com senhas e informações sensíveis, é essencial considerar aspectos éticos e de segurança. Para fortalecer a segurança e prevenir ataques de usuários com mais conhecimento técnico, deve-se focar em assegurar a segurança das senhas em todos os âmbitos, no momento de trafegar dados dentro do *Wi-Fi*, mesmo sendo um AP limitado, no armazenando delas na memória interna do ESP32 e planejar uma geração válida de backup das configurações e das senhas.

6 Conclusão

Ao longo deste trabalho foi desenvolvido um protótipo de um gerenciador de senhas, seguindo a proposta de ter um produto com mais segurança e confiabilidade no gerenciamento de dados via armazenamento das senhas e usuários de forma digital em um hardware desconectado da rede.

O protótipo gerado demonstrou um funcionamento compatível com os objetivos iniciais, apresentando uma potencial aplicação futura, o protótipo ficou com um tamanho compacto e foi utilizado apenas uma porta USB tanto para a emulação do teclado, quanto para a alimentação do circuito.

Sua adoção para uso em condições reais fica condicionada à implementação de algumas características que não foram implementadas durante este projeto, devido a falta de tempo hábil, como a autenticação ao entrar na página WEB e antes de realizar a listagem das descrições das senhas na tela LCD, bem como a encriptação dos dados armazenados na memória Flash e do código presente no ESP32.

Dessa forma, conclui-se que foi possível demonstrar a viabilidade da proposta apresentada, ficando sugestões de melhorias necessárias para futuras pesquisas.

Referências Bibliográfica

ALLYSON, A; LAKSHMI, V Jothi; PACKIALATHA, A. **MOBILE DEVICES USING NFC IN PAYMENT APPLICATIONS**. v. 3. [S.l.], 2015. P. 32–36. ARDUINO. **Arduino Micro**.

[S.l.: s.n.], 2023. Disponível em: <https://docs.arduino.cc/hardware/micro>. BABIUCH, Marek; FOLTÝNEK, Petr; SMUTNÝ, Pavel. Using the ESP32 Microcontroller

for Data Processing. In: p. 1–6. https://www.researchgate.net/profile /Marek-Babiuch/publication/334572715_Using_the_ESP32_Microcontroller_for_Data_P rocessing/links/5e6624eb92851c7ce0536611/Using-the-ESP32-Microcontroller-for -Data-Processing.pdf. DOI: 10.1109/CarpathianCC.2019.8765944. Disponível em: <https://ieeexplore.ieee.org/abstract/document/8765944>. COSKUN, Vedat; OZDENIZCI, Busra;

OK, Kerem. **A survey on near field communication (NFC) technology**. v. 71. [S.l.], ago. 2013. P. 2259–2294. DOI: 10.1007/s11277-012-0935-5. ESWAR, P V D S. **MICRO-**

CONTROLLER MANIPULATED AS HUMAN INTERFACE DEVICE PERFORMING KEYSTROKE INJECTION ATTACK. [S.l.], 2021. P. 1230–1233. Disponível em: <https://www.irjmets.com/uploadedfiles/paper/volume3/issue%5C_7%5C_july%5C_2021/14885/1628083578.pdf>. EXPRESSIF. **Datasheet: ESP32S2 Se-**

ries - Versão1.5. [S.l.], dez. 2022. Disponível em: <https://www.espressif.com/sites/default/files/documentation/esp32-s2%5C_datasheet%5C_en.pdf>. EXPRESSIF. **eFuse Manager**. [S.l.: s.n.], 2023. Disponível em: <https://docs.

espressif.com/projects/esp-idf/en/latest/esp32/api-reference/system/efuse.html>. LI, Zhiwei et al. The Emperors New Password Manager: Secu-

rity Analysis of Web-based Password Managers. In: p. 465–479. ISBN 978-1-931971-15-7. Disponível em: <https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/li%5C_zhiwei>. MOHAMMADIN-

ODOUSHAN, Mohammad et al. Reliable, Secure, and Efficient Hardware Implementation of Password Manager System Using SRAM PUF. **IEEE Access**, v. 9, p. 155711–155725, mai. 2021. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9622 268. ISSN 2169-3536. DOI: 10.1109/ACCESS.2021.3129499. Disponível em: <https://ieeexplore.ieee.org/document/9622%20268>. MYNATT, Elizabeth D. et al. **CHI 2010 : we**

are HCI : conference proceedings, Atlanta, Ga, USA, April 10-15, 2010. [S.l.]: Association for Computing Machinery, 2010. P. 2642. ISBN 9781605589299. Disponível em: <https://dl.acm.org/doi/abs/10.1145/1753326.1753384>. OESCH, Sean; RUOTI,

Scott. That was then, this is now: A security evaluation of password generation, storage, and autofill in browser-based password managers. In: p. 2165–2182. Disponível em: <https://www.usenix.org/system/files/sec20-oesch%5C_0.pdf>. STOBERT,

Elizabeth; BIDDLE, Robert. A Password Manager That Doesn't Remember Passwords. In: p. 39–52. https://www.nspw.org/papers/2014/nspw2014-stobert.pdf. ISBN 9781450330626. DOI: 10.1145/2683467.2683471. Disponível em: <https://doi.org/10.1145/2683467.2683471>. VIEW, Georgia Webbsit; SUMMERS, Wayne C; BOSWORTH, Ed-

ward. **Password policy: The good, the bad, and the ugly**. [S.l.], 2004. Disponível em: <https://www.researchgate.net/profile/Wayne-Summers-2/publication/234799064%5C_Password%5C_policy%5C_The%5C_good%5C_the%5C_bad%5C_and%5C_the%5C_ugly/links/54f204310cf2f9e34eff3d50/Password-policy-The-good-the-bad-and-the-ugly.pdf>. WATELECTRONICS. **What is**

LCD 16X2 : Pin Configuration and Its Working. [S.l.: s.n.], ago. 2021.