

BOTÃO DE SEGURANÇA: BOTÃO DE SEGURANÇA PARA SALVAGUARDA HOSPITALAR E AMBULATORIAL

*SECURITY BUTTON: SECURITY BUTTON FOR HOSPITAL AND OUTPATIENT
SAGEGUARD*

Leandro C. Lima¹, Emerson A. Mouco Júnior²

¹Faculdade de Tecnologia Prof. José Camargo – Fatec Jales, leandroo.cl@hotmail.com

²Faculdade de Tecnologia Prof. José Camargo – Fatec Jales, emerson.mouco@fatec.sp.gov.br

Informação e Comunicação

Subárea: Tecnologia da Informação

RESUMO

Devido aos relatos de ameaças e agressões a profissionais médicos em ambientes ambulatoriais, surgiu a necessidade da implantação de um sistema de emergência no qual o profissional pode acionar um botão físico, emitindo assim um pedido de ajuda para a equipe de segurança local. O sistema foi desenvolvido utilizando uma combinação de tecnologias e linguagens de programação, sendo estas: Arduino, HTML, CSS, Protocolos de Rede, Javascript, AJAX e C++. O Arduino desempenha um papel central na funcionalidade do sistema, permitindo que o botão físico seja conectado à rede Wi-Fi e acione um alerta de emergência instantaneamente. Essas tecnologias se complementam para criar uma solução prática, confiável e ágil para lidar com situações de emergência. Esse sistema demonstrou ser mais eficaz e ágil do que alternativas baseadas em software, garantindo uma resposta rápida em situações críticas. A implementação do “Botão de Segurança para Salvaguarda Hospitalar e Ambulatorial” é uma medida essencial para melhorar a segurança no ambiente de consultórios médicos, proporcionando tranquilidade aos profissionais de saúde e aos pacientes.

Palavras-chave: botão de segurança; salvaguarda; emergência.

ABSTRACT

Due to reports of threats and assaults against medical professionals in outpatient settings, there arose a need for the implementation of an emergency system where the professional can activate a physical button, thus issuing a call for help to the local security team. The system was developed using a combination of technologies and programming languages, including: Arduino, HTML, CSS, Network Protocols, Javascript, AJAX, and C++. Arduino plays a central role in the functionality of the system, allowing the physical button to be connected to the Wi-Fi network and trigger an emergency alert instantly. These technologies complement each other to create a practical, reliable, and swift solution for handling emergency situations. This system has proven to be more effective and responsive than software-based alternatives, ensuring a prompt response in critical situations. The implementation of the “Security Button for Hospital and Outpatient Safeguard” is an essential measure to enhance security in medical office environments, providing peace of mind for healthcare professionals and patients.

Keywords: security button; safeguard; emergency.

1 INTRODUÇÃO

A crescente preocupação com a segurança nos consultórios médicos é impulsionada por incidentes recorrentes, evidenciados em reportagens e pesquisas que destacam ameaças e agressões direcionadas a profissionais de saúde durante o exercício de suas atividades. A gravidade desses eventos é destacada em estudos como os disponibilizados em fontes

confiáveis, como o trabalho realizado pela Faculdade de Medicina da Universidade Federal de Minas Gerais (UFMG), que aborda a violência em serviços de saúde (Silva Júnior; Oliveira, 2020). Além disso, o Conselho Regional de Enfermagem de São Paulo (COREN-SP) também fornece insights valiosos sobre a segurança no ambiente de saúde (Baptista *et al.*, 2017). Outros relatos, como o ocorrido em um posto de saúde, conforme registrado pelo portal Estado de Minas, e a pesquisa conduzida pelo portal UOL, que destaca que 7 em cada 10 médicos sofrem agressões, ameaças e xingamentos no ambiente de trabalho, sublinham a urgência de abordar essa questão (Preite Sobrinho, 2018). Profissionais da saúde em Salvador também denunciaram ameaças de pacientes, como documentado pela matéria do G1, demonstrando a complexidade e a abrangência do desafio que este projeto se propõe a explorar (Profissionais..., 2022). Dessa forma, a análise cuidadosa desses incidentes e a busca por soluções tornam-se essenciais não apenas para compreender a gravidade do problema, mas também para prevenir a escalada dessas situações para eventos mais graves.

Estabelecer estratégias eficazes de segurança não apenas responde imediatamente a ameaças e agressões, mas também se configura como uma sólida salvaguarda, proporcionando um ambiente mais seguro e protegido para os profissionais de saúde. Visando mitigar tais riscos e assegurar a integridade física dos médicos e demais membros da equipe médica, foi desenvolvido o "Botão de Segurança para Salvaguarda Hospitalar e Ambulatorial". Este sistema consiste em um botão físico, conectado a um Arduino alimentado por pilhas e com acesso ao Wi-Fi, que permite aos médicos solicitar assistência aos seguranças em tempo real, caso se sintam ameaçados durante o atendimento a pacientes.

A implementação do "Botão de Segurança para Salvaguarda Hospitalar e Ambulatorial" é realizada através da utilização de tecnologias como o Arduino e linguagens de programação como HTML, JavaScript, AJAX e C++. O sistema proporciona uma forma rápida e eficaz de comunicar a ocorrência de uma situação de emergência à equipe de segurança. Quando o botão de emergência é acionado, um alerta é instantaneamente transmitido ao computador da guarita, onde os seguranças monitoram as câmeras e recebem informações importantes para agir prontamente e prevenir incidentes mais graves.

A relevância deste trabalho está na melhoria da segurança no ambiente de consultórios médicos, que é fundamental para preservar a integridade desses profissionais, evitando com que situações de tensão possam se agravar sem um pedido de ajuda imediato, garantindo a continuidade dos serviços de saúde e preservando ambas as partes. Além disso, o custo relativamente baixo de implementação do sistema é uma justificativa adicional para sua adoção generalizada. Com um investimento de aproximadamente cinquenta reais por unidade do botão de emergência e a utilização de recursos de computação já disponíveis na guarita, a solução pode oferecer um alto retorno em termos de segurança e tranquilidade para os profissionais médicos.

2 REFERENCIAL TEÓRICO

A segurança no ambiente de consultórios médicos é de extrema importância, considerando os crescentes relatos de incidentes, vistos nas reportagens acima, que reforçam a existência de ameaças e agressões a profissionais da saúde durante o atendimento. Estudos mostram que incidentes dessa natureza têm implicações sérias na vida desses profissionais. De acordo com Cezar (2005), em seu artigo "Problemas de violência ocupacional em um serviço de urgência hospitalar da Cidade de Londrina, Paraná, Brasil", a violência no trabalho tem se apresentado como um fator preocupante em muitos hospitais de todo o mundo. A Organização Mundial da Saúde (OMS) também discorre a respeito da violência. Para a OMS a violência é a utilização deliberada da força física ou do poder, seja concretamente aplicada ou através de

ameaças, dirigida a uma pessoa, grupo ou comunidade, podendo resultar em morte, lesões, impacto psicológico, alterações no desenvolvimento ou privação.

Para o desenvolvimento de tal sistema, composto por hardware e software é necessário o conhecimento em diversas áreas como linguagens de programação, criação, eletrônica básica/arduino e redes de computadores. A complexidade do empreendimento requer uma abordagem integrada que abrange desde a escrita de código em linguagens como C++, JavaScript e a manipulação de tecnologias web, como HTML, CSS e AJAX, até o domínio de protocolos de rede essenciais para a comunicação eficiente entre os diferentes componentes do sistema. Falaremos a seguir sobre as tecnologias utilizadas no projeto.

2.1 ARDUINO

McRoberts (2015) descreve o Arduino em seu livro, “Arduino Básico”, como um dispositivo compacto e programável que viabiliza o processamento de informações provenientes de entradas e a saída de dados para componentes externos conectados a ele. Esse dispositivo é categorizado como uma plataforma física ou embarcada, ou seja, um sistema capaz de interagir com o ambiente, seja por meio de componentes de hardware ou por software. Segundo McRoberts (2015), o Arduino tem a capacidade de coletar dados de sensores e transmiti-los para um site, permitindo a exibição desses dados em formato gráfico.

A origem dessa notável plataforma remonta a 2005, quando surgiu na cidade de Ivrea, Itália, sob a liderança de Massimo Banzi. O Arduino foi concebido em resposta aos elevados custos das plataformas de prototipagem disponíveis naquela época, que limitavam o acesso ao conhecimento de muitas pessoas. Portanto, a missão primordial do Arduino era reduzir o custo dessas plataformas para torná-las acessíveis a escolas e facilitar a aprendizagem de um público mais amplo.

Um aspecto notável do Arduino é a sua natureza de hardware aberto, o que significa que qualquer pessoa com um mínimo de conhecimento em eletrônica, ou mesmo sem conhecimento algum, pode personalizar o layout do seu Arduino, acrescentando ou removendo componentes, entre outras modificações. Tanto o hardware quanto o software do Arduino são de código aberto, permitindo que qualquer pessoa utilize livremente o código, os esquemas e o projeto para qualquer finalidade. Essa abertura resultou na disponibilidade de inúmeras placas-clone e outras variantes baseadas no Arduino, que podem ser adquiridas no mercado ou fabricadas a partir de diagramas (McRoberts, 2015).

Essa flexibilidade de customização do hardware do Arduino levou ao surgimento de diversas variações no mercado. Para o desenvolvimento da conexão entre dispositivos e um servidor web em um projeto, foi considerada a utilização de uma placa Arduino devido ao seu baixo custo e documentação amplamente disponível para construção. A escolha do Arduino também se baseou na sua facilidade de programação, que utiliza principalmente as linguagens C e C++, além de oferecer um ambiente de programação gratuito. O custo acessível torna o Arduino uma escolha popular em diversas áreas, especialmente na robótica e na Internet das Coisas (IoT), onde é utilizado para conectar dispositivos à web. O Arduino permite a integração de sensores para monitorar o ambiente, permitindo a construção de placas personalizadas de acordo com as necessidades do projeto.

Além disso, surgiram várias adaptações do Arduino com processadores eficientes de baixo consumo de energia e portas de expansão tanto analógicas quanto digitais. No contexto do projeto em questão, uma placa Arduino compatível com USB é utilizada na maioria das conexões entre dispositivos e computadores.

A ideia é fazer com que o Arduino, através de um webserver, exiba em tempo real uma página contendo informações da sua localidade, como por exemplo “CONSULTÓRIO 30” em uma página HTML com fundo branco. Ao ser pressionado um Push Button (Chave Táctil) é

executado uma função em Java Script que altera o estado da página previamente visualizada em tempo real no computador onde se faz o monitoramento, mudando seu fundo de branco para vermelho e executando um player nativo do HTML com um áudio de alerta, este por sua vez embutido na memória flash do Arduino, durando o tempo que for desejado.

2.2 HTML

De acordo com Robson e Freeman (2015), o HTML (*HiperText Markup Language*) é uma linguagem que se destina à criação de hiperlinks, proporcionando uma maneira de estruturar o texto por meio de *tags*, que comunicam ao navegador a estrutura do documento. No contexto do HTML, as *tags* são utilizadas para informar ao navegador qual é o tipo de documento em questão. Os autores destacam que é dever do HTML fornecer todas as informações necessárias ao navegador para que ele possa renderizar a página da web. Eles também enfatizam que a criação de um arquivo HTML requer apenas um texto simples e descomplicado.

O HTML é uma linguagem especializada voltada para a apresentação e acesso de páginas da web. Trata-se de uma linguagem textual comum que utiliza códigos conhecidos como comandos da linguagem. Destaca-se os principais elementos da linguagem HTML, que incluem títulos, textos, parágrafos, imagens e links. Esses elementos desempenham um papel fundamental na exibição de outras páginas na tela. A ideia é que no computador, onde se faz o monitoramento, tenha uma página HTML contendo diversas *tags iframe*, cujo seus respectivos links se refiram à uma localidade onde se encontra o botão instalado, ficando sempre visível em tempo real, permitindo assim que um eventual mal funcionamento em qualquer um dos dispositivos seja visualizado na tela.

2.3 JAVASCRIPT

Segundo as observações de Ramalho (2001), o uso exclusivo da linguagem HTML não proporciona a capacidade de criar animações e efeitos especiais para tornar uma página interativa. Para atingir esse objetivo, é essencial combinar o HTML com o JavaScript. O autor enfatiza que é fundamental entender a distinção entre as linguagens Java e JavaScript. Java é uma linguagem de aplicação independente, especializada na criação de pequenos programas distribuídos na internet, enquanto o JavaScript é uma linguagem que opera dentro de um programa HTML. Inicialmente denominada Live Script, a linguagem adotou o nome JavaScript por motivos de marketing.

Ramalho (2001) ainda descreve o JavaScript como uma linguagem que é interpretada em tempo real, sendo traduzida para uma linguagem compreensível pelo computador no momento da execução. Essa linguagem é baseada em objetos, que podem ser objetos pré-definidos ou criados pelo programador. Além das variáveis que armazenam informações sobre objetos, o JavaScript permite a criação de variáveis de memória temporárias que podem armazenar datas, valores e texto. Após uma breve compreensão dessas linguagens, é importante notar que a utilização do CSS (*Cascading Style Sheets*) desempenha um papel relevante, uma vez que estrutura o código e proporciona maior interatividade à página.

No contexto deste projeto, as funções em Java Script fazem com que seja alterado o estado da página para ser visualizada em tempo real e fica responsável por retornar ao estado anterior após determinado tempo previamente definido. De forma dinâmica a página faz a interpretação do código enviado pelo microcontrolador, sem depender de programas instalados localmente, trazendo assim a possibilidade de ser executado em qualquer dispositivo que tenha apenas um navegador instalado em seu sistema operacional, como um computador, tablet ou celular.

2.4 CSS – Cascading Style Sheets

Conforme Meyer (2006), o CSS (*Cascading Style Sheets*) surgiu como uma linguagem crucial para a apresentação de páginas da web. Ele proporciona o controle sobre a estilização e o *layout* das páginas, permitindo que os desenvolvedores definam a aparência visual de seus *sites*. Meyer (2006) destaca que o CSS é fundamental para separar a estrutura do conteúdo HTML da sua apresentação, o que torna as páginas mais flexíveis e fáceis de serem estilizadas.

2.5 AJAX

De acordo com Riordan (2009), o Ajax revolucionou o desenvolvimento web, permitindo a criação de aplicações web mais eficientes e responsivas. Ela destaca como o Ajax possibilita que os aplicativos web solicitem e recebam dados do servidor sem recarregar a página inteira. Essa tecnologia é essencial para criar experiências de usuário mais dinâmicas e interativas na web. Isso garante que nosso sistema funcione dinamicamente, sem a necessidade de intervenção do usuário atualizar a página manualmente onde se localiza o monitoramento dos dispositivos.

2.6 C++

Segundo Souza (2018), o C++ é uma linguagem de programação de baixo nível em comparação com outras linguagens de alto nível como o Java, sendo amplamente utilizada, devido à sua eficiência, performance e flexibilidade. Ele enfatiza que o uso do C++ permite que os desenvolvedores aproveitem ao máximo o hardware, escrevendo código que seja otimizado para as tarefas específicas que desejam realizar. Além disso, o C++ oferece suporte à programação orientada a objetos, o que facilita a organização do código para projetos mais complexos. Esta incrível linguagem nos proporciona o total controle e personalização de nosso sistema baseado em Arduino, com desempenho e confiabilidade.

2.7 PROTOCOLOS DE REDE E WI-FI

Segundo Tanenbaum (2011), os protocolos de rede desempenham um papel fundamental na comunicação de dispositivos em redes, como o TCP/IP. A arquitetura TCP/IP é um conjunto de protocolos que governa a comunicação na Internet e em muitas redes locais, incluindo redes Wi-Fi. Tanenbaum (2011), destaca que o TCP/IP é a base da comunicação na Internet, garantindo a entrega confiável de dados entre dispositivos em redes IP. Ele ressalta que o TCP (*Transmission Control Protocol*) é um dos protocolos mais importantes da pilha TCP/IP, sendo responsável pela transmissão confiável de dados entre dispositivos. Além disso, o IP (*Internet Protocol*) cuida do roteamento e entrega de pacotes de dados, permitindo a conectividade na Internet e em redes locais.

Em relação às redes Wi-Fi, Tanenbaum (2011) menciona que o Wi-Fi é uma tecnologia de comunicação sem fio que utiliza o padrão IEEE 802.11. Ele destaca a importância de entender os protocolos subjacentes que permitem a comunicação sem fio, bem como as questões de segurança relacionadas a redes Wi-Fi, como a criptografia e a autenticação. No cenário em questão, os dispositivos conectam-se individualmente em Access Points (Pontos de Acesso) e trafegam dados utilizando o protocolo TCP/IP, servindo a página HTML onde os eventos de cada botão são monitorados de maneira individual.

3 METODOLOGIA

Nesta seção, será descrito como os testes práticos foram conduzidos para avaliar a eficácia do "Botão de Segurança para Salvaguarda Hospitalar e Ambulatorial" em uma instituição de saúde na cidade de Jales.

3.1 TESTES PRÁTICOS COM SOFTWARE (FASE INICIAL)

- Inicialmente, foram realizados testes práticos em um ambiente de software instalado nos computadores de consultórios médicos. O objetivo desses testes era avaliar a eficácia do sistema de botão de segurança baseado em software em situações simuladas de potencial ameaça ou violência;

- Durante um período de teste de um mês, os médicos participantes usaram o software para acionar um alarme de emergência em situações simuladas;

- Foi registrado que o tempo médio desde o início do incidente até o envio do alerta foi de aproximadamente 35 segundos, sendo 10 segundos destinados à localização do software e à execução do clique duplo, 15 segundos para a execução do software e 10 segundos para a locomoção da equipe de segurança;

- Prós do Software: O software oferece uma solução econômica em termos de custos de implementação. No entanto, durante os testes, observamos que o software é menos prático e confiável, uma vez que depende do sistema operacional do computador, que veio a travar durante um acionamento, ocasionando uma falha. Além disso, em situações de tensão ou perigo, a localização e execução do software podem se tornar inviáveis.

3.2 TESTES COM BOTÃO FÍSICO (SEGUNDA FASE)

- Com base nos resultados da fase inicial dos testes foi decidido implementar o "Botão de Segurança" em um único consultório médico para fins de comparação. O botão físico foi posicionado em um local de fácil acesso;

- Durante um período de teste de um mês, o autor realizou o acionamento do botão físico para emitir um alerta de emergência em situações simuladas;

- Foi registrado que o tempo médio desde o início do hipotético incidente até o envio do alerta foi em torno de 10 segundos, com o alerta sendo visualizado instantaneamente, na interface de monitoramento;

- Prós do Botão Físico: O botão físico demonstrou ser mais prático e fácil de acionar em emergências, pois não depende do funcionamento do sistema operacional do computador. Além disso, é mais confiável, uma vez que não está sujeito a falhas de software ou problemas de eletricidade que podem afetar a operação do software.

3.3 ANÁLISE DE DADOS COMPARATIVOS

Foram realizadas análises dos tempos registrados durante as duas fases de testes para comparar a eficácia do sistema baseado em software e do botão físico em termos de tempo de resposta, como discutiremos a seguir.

3.4 RESULTADOS ESPERADOS

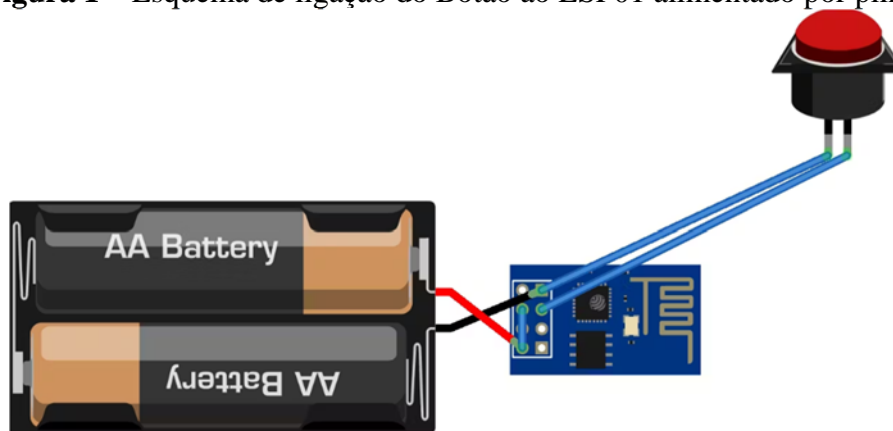
Com base nos dados dos testes, espera-se que os resultados demonstrem claramente que o "Botão de Segurança" possui uma eficácia superior em proporcionar uma resposta rápida a potenciais situações de ameaça ou violência nos consultórios médicos. Essa comparação deve

auxiliar na determinação de qual solução é mais confiável e eficaz na melhoria da segurança no ambiente de atendimento médico. A metodologia de testes práticos adotada possibilitou uma avaliação direta da funcionalidade tanto do sistema baseado em software quanto do botão de emergência físico, com resultados práticos que sustentam a superioridade do botão físico em termos de tempo de resposta e praticidade.

4 ANÁLISE E DISCUSSÃO DOS RESULTADOS

Durante a condução dos testes práticos, foram observadas diferenças significativas nos tempos de resposta entre o sistema baseado em software e o botão de segurança físico. Esses resultados têm implicações diretas na eficácia da resposta a potenciais situações de ameaça ou violência nos consultórios médicos, como veremos a seguir.

Figura 1 – Esquema de ligação do Botão ao ESP01 alimentado por pilhas



Fonte: Elaborada pelos autores.

4.1 COMPARAÇÃO DE TEMPOS

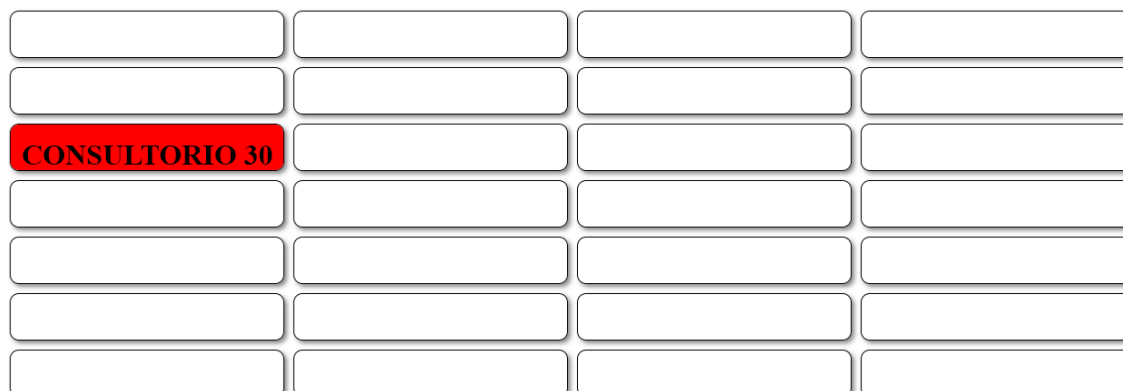
No cenário de software o tempo médio desde o início do incidente até o envio do alerta foi de 35 segundos. Desses, 10 segundos foram dedicados à localização do software, 15 segundos para a execução e mais 10 segundos para uma eventual locomoção da equipe de segurança. Por outro lado, a implementação do botão de segurança físico demonstrou tempos consideravelmente mais rápidos, com uma média de 10 segundos.

Figura 2 – Tela de monitoramento constando um dispositivo ativo no CONSULTÓRIO 30

SISTEMA DE SEGURANÇA			
CONSULTORIO 30			

Fonte: Elaborada pelos autores.

Figura 3 – Tela de monitoramento com alerta disparado no CONSULTÓRIO 30
SISTEMA DE SEGURANÇA



Fonte: Elaborada pelos autores.

4.2 IMPORTÂNCIA DOS SEGUNDOS

A diferença de 25 segundos entre o tempo de resposta do software e o botão de segurança físico é crucial em situações de potencial ameaça ou violência. Esses poucos segundos podem fazer a diferença entre uma ação bem-sucedida de prevenção e uma situação mais séria e perigosa.

Em situações de tensão, é vital que a equipe de segurança seja notificada o mais rápido possível para tomar as medidas apropriadas. A demora na localização e execução do software pode atrasar essa notificação, permitindo que a situação se agrave antes que a ajuda chegue.

4.3 CONCLUSÕES E IMPLICAÇÕES

Com base na análise dos resultados, fica claro que a implementação do botão de segurança físico oferece uma vantagem significativa em termos de tempo de resposta em comparação com o sistema baseado em software. Isso é fundamental para a prevenção de incidentes graves nos consultórios médicos.

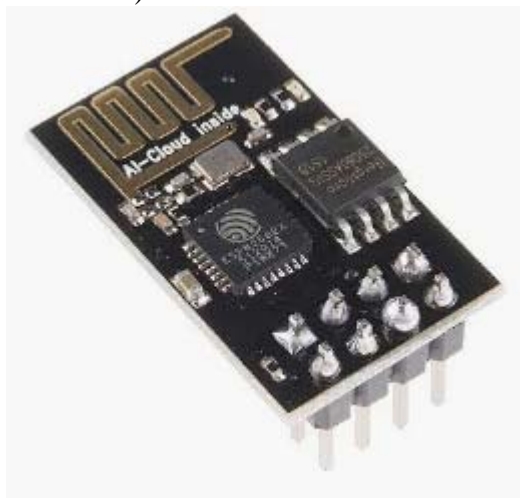
A implementação de um botão físico é mais prática, fácil de acionar e confiável, uma vez que não depende do funcionamento do sistema operacional do computador, além do baixo custo de implantação como mostra a Tabela 1 abaixo.

Tabela 1 – Média de custos por componente

Componente	Preço médio
ESP01	R\$21,76
Push Button	R\$ 6,56
Suporte para pilhas AA	R\$ 4,66
Total	R\$ 32,98

Fonte: Elaborada pelos autores.

Figura 4 – ESP01 (Microcontrolador similar ao Arduino, com as mesmas funcionalidades, Wi-Fi embutido e custo reduzido)



Fonte: Elaborada pelos autores.

Figura 5 – Push Button (Chave Táctil)



Fonte: Elaborada pelos autores.

Figura 6 – Suporte para pilhas AA

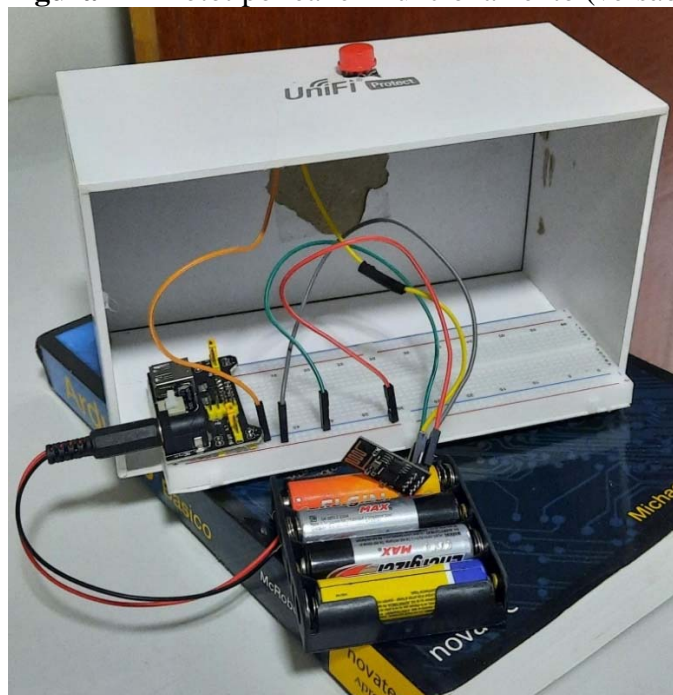


Fonte: Elaborada pelos autores.

Os resultados destacam a importância de considerar não apenas a eficácia geral do sistema, mas também a velocidade de resposta em situações críticas. Essa análise reforça a necessidade de adotar soluções de segurança mais ágeis e eficientes, como o botão de segurança

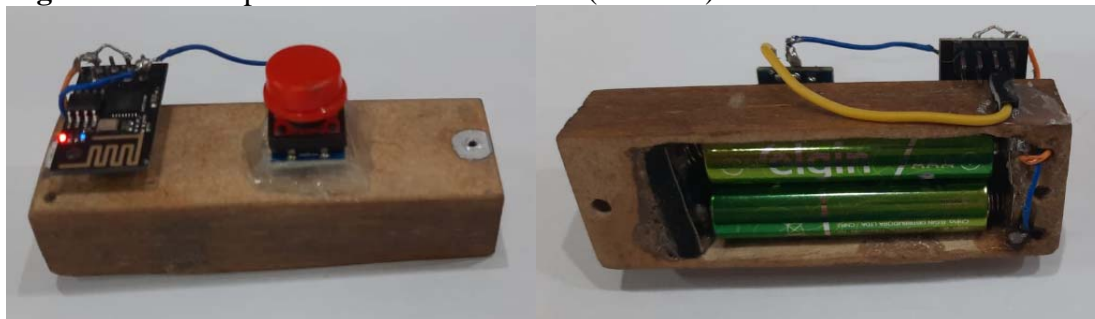
físico, para garantir a salvaguarda adequada dos profissionais médicos e dos pacientes em ambientes hospitalares e ambulatoriais.

Figura 7 – Protótipo real em funcionamento (versão 1)



Fonte: Elaborada pelos autores.

Figura 8 – Protótipo real em funcionamento (versão 2)



Fonte: Elaborada pelos autores.

Figura 9 – Código embutido no ESP01 (dispositivo)

```
1 #include <ESP8266WiFi.h> // Inclui a biblioteca ESP8266 para suporte Wi-Fi
2 #include <ESPAsyncTCP.h> // Inclui a biblioteca AsyncTCP para comunicação assíncrona
3 #include <ESPAsyncWebServer.h> // Inclui a biblioteca AsyncWebServer para criação do servidor web
4 #include <LittleFS.h> // Inclui a biblioteca LittleFS para manipulação do sistema de arquivos
5
6 const char* ssid = "MELFINET_FIBRA - ROSSINI"; // Nome da rede Wi-Fi
7 const char* password = "*****"; // Senha da rede Wi-Fi
8 IPAddress staticIP(192, 168, 0, 120); // Endereço IP estático
9 IPAddress gateway(192, 168, 0, 1); // Endereço do gateway
10 IPAddress subnet(255, 255, 255, 0); // Máscara de sub-rede
11
12 AsyncWebServer server(80); // Cria um servidor web assíncrono na porta 80
13
14 bool buttonPressed = false; // Variável para rastrear o estado do botão
15 unsigned long lastButtonPress = 0; // Variável para rastrear o tempo da última pressão do botão
16
17 void setup() {
18 // Função de inicialização do Arduino
19
20 WiFi.begin(ssid, password); // Inicia a conexão Wi-Fi com as credenciais definidas
21 WiFi.config(staticIP, gateway, subnet); // Configura o endereço IP estático
22
23 while (WiFi.status() != WL_CONNECTED) {
24 delay(1000); // Aguarda até que a conexão Wi-Fi seja estabelecida
25 }
26
27 pinMode(0, INPUT_PULLUP); // Configura o pino 0 como entrada com resistor pull-up interno
28
29 if (!LittleFS.begin()) {
30 // Se falhar ao montar o sistema de arquivos LittleFS
31 // Serial.println("Failed to mount LittleFS"); // Mensagem de depuração
32 return; // Retorna e não continua a configuração
33 }
34
35 if (!LittleFS.exists("/alarme.wav")) {
36 // Se o arquivo de áudio "alarme.wav" não for encontrado no sistema de arquivos
37 // Serial.println("Audio file not found"); // Mensagem de depuração
38 return; // Retorna e não continua a configuração
39 }
40
41 // Configuração das rotas do servidor web
42 server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){
43 int buttonState = digitalRead(0); // Lê o estado do pino 0 (presumivelmente um botão)
44
45 if (buttonState == LOW && !buttonPressed) {
46 buttonPressed = true;
47 lastButtonPress = millis();
48 }
49
50 if (buttonPressed && (millis() - lastButtonPress <= 10000)) {
51 // Se o botão foi pressionado e o tempo desde a última pressão é menor ou igual a 10 segundos
52 String redPage = "<html><head><style>body { background-color: red; display: flex; flex-direction: column; justify-content: center; align-items: center; height: 100vh; }</style>";
53 redPage += "<script>setTimeout(function() { location.reload(); }, 10000);</script></head><body><h1>CONSULTORIO</h1>";
54 redPage += "<audio id='alarmAudio' autoplay loop><source src='alarme.wav' type='audio/wav'>Seu navegador não suporta o elemento de áudio.</audio>";
55 redPage += "</body></html>";
56 request->send(200, "text/html", redPage); // Envia uma página HTML vermelha
57 } else {
58 buttonPressed = false;
59 String whitePage = "<html><head><style>body { display: flex; flex-direction: column; justify-content: center; align-items: center; height: 100vh; }</style>";
60 whitePage += "<script>setTimeout(function() { location.reload(); }, 1000);</script></head><body><h1>CONSULTORIO</h1></body></html>";
61 request->send(200, "text/html", whitePage); // Envia uma página HTML branca
62 }
63 });
64
65 server.on("/alarme.wav", HTTP_GET, [](AsyncWebServerRequest *request){
66 File file = LittleFS.open("/alarme.wav", "r"); // Abre o arquivo de áudio para leitura
67
68 if (file) {
69 request->send(LittleFS, "/alarme.wav", "audio/wav"); // Envia o arquivo de áudio como resposta
70 file.close(); // Fecha o arquivo
71 }
72 });
73
74 server.on("/ajax", HTTP_GET, [](AsyncWebServerRequest *request){
75 String response = (buttonPressed) ? "1" : "0"; // Retorna "1" se o botão foi pressionado, caso contrário, "0"
76 request->send(200, "text/plain", response); // Envia a resposta como texto simples
77 });
78
79 server.begin(); // Inicia o servidor web
80 }
81
82 void loop() {
83 }
```

Fonte: Elaborada pelos autores.

Figura 10 – Código da Página HTML para ser exibida no computador de monitoramento

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>SISTEMA DE SEGURANÇA</title>
7   <style>
8     body {
9       font-family: Arial, sans-serif;
10    }
11    #container {
12      display: flex;
13      flex-wrap: wrap;
14      justify-content: center;
15      align-items: center;
16      height: 10vh;
17      position: relative;
18    }
19    h1 {
20      text-align: center;
21      font-weight: bold;
22    }
23    iframe {
24      width: 300px;
25      height: 50px;
26      border: 1px solid #000;
27      border-radius: 10px;
28      margin: 5px;
29      box-shadow: 3px 3px 5px #888888;
30    }
31  </style>
32 </head>
33 <body>
34   <h1>SISTEMA DE SEGURANÇA</h1>
35   <div id="container">
36     <!-- iframes com os links dos dispositivos -->
37     <iframe src="http://192.168.8.128" scrolling="no"></iframe>
38     <iframe src="http://inserir ip do dispositivo aqui" scrolling="no"></iframe>
39     <iframe src="http://inserir ip do dispositivo aqui" scrolling="no"></iframe>
40   |
41     <!-- Adicione os demais iframes conforme necessário -->
42   </div>
43   <script>
44     // Função para atualizar os iframes a cada 1 segundo
45     function atualizarIframes() {
46       const iframes = document.querySelectorAll("iframe");
47       iframes.forEach(function(iframe) {
48         iframe.src = iframe.src; // Recarrega o iframe
49       });
50     }
51     // Atualize os iframes a cada 1 segundo
52     setInterval(atualizarIframes, 1000);
53   </script>
54 </body>
55 </html>
```

Fonte: Elaborada pelos autores.

5 CONSIDERAÇÕES FINAIS

Atualmente não se encontram soluções objetivas que atendam estas necessidades no mercado nacional. Algumas empresas prestam serviços por assinatura encarecendo a sua viabilidade e algumas instituições realizam por conta própria a implementação de dispositivos que emitem algum tipo de alerta para tentar prevenir ou atender de forma rápida algum pedido de socorro. No entanto, as soluções conhecidas até o momento demandam maior investimento e alterações prediais, com uso de interruptores e cabeamento elétrico para acionamento de

lâmpadas ou campainhas, como parece ser o caso divulgado pela Prefeitura de Caraguatatuba, onde o sistema foi implantado (Caraguatatuba, 2023).

Com base nos resultados dos testes, conclui-se que o "Botão de Segurança" é uma solução mais eficaz em comparação com o sistema baseado em software ou outras opções parecidas disponíveis no mercado. Ele oferece uma resposta mais rápida em situações de ameaça ou violência, tornando-o uma escolha prática e confiável para garantir a segurança nos consultórios médicos. Destacamos diversos diferenciais como o reaproveitamento da infraestrutura Wi-Fi já existente, baixo custo de implementação e tempo de resposta ao acionamento.

Recomenda-se a implementação do "Botão de Segurança" em todas as instituições de saúde, sendo elas clínicas, consultórios, ambulatórios e hospitais como medida de segurança eficaz. Essa solução contribuirá significativamente para a segurança de médicos e demais profissionais de saúde.

REFERÊNCIAS

BAPTISTA, P. C. P. *et al.* **Violência no trabalho**: guia de prevenção para os profissionais de enfermagem. São Paulo: Coren-SP, 2017.

CARAGUATATUBA. Secretaria da Saúde. **Caraguatatuba cria botão do pânico em consultórios para salvaguardar integridade de profissionais de saúde**. 2023. Disponível em: <https://www.caraguatatuba.sp.gov.br/pmc/2023/08/caraguatatuba-cria-botao-do-panico-em-consultorios-para-salvaguardar-integridade-de-profissionais-de-saude/>. Acesso em: 31 out. 2023.

CEZAR, E. S. **Problemas de violência ocupacional em um serviço de urgência hospitalar da cidade de Londrina-Paraná**. 2005. Dissertação (Mestrado em Enfermagem Fundamental) – Universidade de São Paulo, Ribeirão Preto, 2005. Disponível em: https://teses.usp.br/teses/disponiveis/22/22132/tde-06052005-213359/publico/Cezar_ES.pdf. Acesso em: 31 out. 2023.

MCRBERTS, S. **Arduino básico**. 2. ed. São Paulo: Novatec, 2015.

MEYER, E. A. **CSS: the definitive guide**. 3. ed. Califórnia, O'Reilly, 2006.

PREITE SOBRINHO, W. **Em SP, 7 em casa 10 médicos sofrem agressões, ameaças e xingamentos**. 2018. Disponível em: <https://noticias.uol.com.br/saude/ultimas-noticias/redacao/2018/09/13/7-em-cada-10-medicos-sofrem-agressoes-ameacas-e-xingamentos-no-trabalho.htm>. Acesso em: 30 out. 2023.

PROFISSIONAIS da saúde denunciam ameaças de pacientes em posto de Salvador, 'descontrole total', avalia Leo Prates. **G1 Bahia**, 19 jan. 2022. Disponível em: <https://g1.globo.com/ba/bahia/noticia/2022/01/19/profissionais-da-saude-denunciam-ameacas-de-pacientes-em-posto-de-salvador-descontrole-total-avalia-leo-prates.ghtml>. Acesso em: 30 out. 2023.

RAMALHO, J. A. **Java script: prático e rápido**. 2. ed. São Paulo: Berkeley, 2001.

RIORDAN, R. M. **Ajax profissional**. Rio de Janeiro: Alta Books, 2009. (Use a cabeça!).

ROBSON, E.; FREEMAN, E. **HTML e CSS**. Rio de Janeiro: Alta Books, 2015. (Use a cabeça!).

SILVA JÚNIOR, J. A.; OLIVEIRA, L. S. Violência em serviços de saúde. **Jornada da disciplina saúde do trabalhador**, 2020. Disponível em: https://www.medicina.ufmg.br/wp-content/uploads/sites/7/2020/11/inserir_-Viol%C3%Aancia-em-servi%C3%A7os-de-sa%C3%BAde_anexo-14_2020.pdf. Acesso em: 30 out. 2023.

SOUZA, M. **Dominando C++**: curso completo. 2018. Disponível em: <https://hotmart.com/pt-br/marketplace/produtos/dominando-c-curso-completo/L83406464S>. Acesso em: 31 out. 2023.

TANENBAUM, A. S. **Redes de computadores**. 5. ed. São Paulo: Pearson, 2011.