REVISTA CIENTÍFICA - RECIMA21 ISSN 2675-6218

SEGURANÇA DIGITAL ACESSÍVEL: UMA SOLUÇÃO DE VERIFICAÇÃO MANUAL DE LINKS SUSPEITOS

AFFORDABLE DIGITAL SECURITY: A MANUAL SOLUTION FOR CHECKING SUSPICIOUS LINKS

SEGURIDAD DIGITAL ACCESIBLE: UNA SOLUCIÓN DE VERIFICACIÓN MANUAL DE ENLACES SOSPECHOSOS

Christian Kevelyn da Silva[1], Fabiana Florian[2], João Henrique Gião Borges[3]

https://doi.org/10.47820/recima21.v6i1.7052

PUBLICADO: 11/2025

RESUMO

Este trabalho tem como objetivo desenvolver uma ferramenta prática e acessível voltada à segurança digital, capaz de auxiliar usuários na verificação manual de links suspeitos. A proposta visa reduzir a exposição a fraudes *online*, especialmente aquelas disseminadas por meio de redes sociais. Foram realizadas pesquisas bibliográficas com foco no levantamento de dados sobre táticas de fraude digital e práticas de segurança da informação. Em seguida, foi implementada uma solução utilizando linguagens e *frameworks* como JavaScript, Python e SQLite. O sistema foi testado com URLs legítimas e fraudulentas, analisando parâmetros como uso de HTTPS, validade de certificados SSL e data de criação do domínio. Os resultados demonstraram que a ferramenta cumpre sua função ao classificar adequadamente os sites em níveis de risco (“seguro”, “moderado” e “suspeito”), oferecendo ao usuário uma visualização clara e imediata da confiabilidade das páginas acessadas. A integração entre os módulos mostrou-se estável e eficiente. Conclui-se que a solução proposta contribui para a democratização da segurança digital, ao oferecer uma camada adicional de proteção e conscientização durante a navegação.

PALAVRAS-CHAVE: Segurança digital. Links suspeitos. Extensão de navegador. *Flask*. Fraude *online*.

ABSTRACT

This paper aims to develop a practical and accessible tool focused on digital security, capable of assisting users in manually verifying suspicious links. The proposal seeks to reduce exposure to online fraud, especially those spread through social media. Bibliographic research was conducted with a focus on gathering data on digital fraud tactics and information security practices. Subsequently, a solution was implemented using languages and frameworks such as JavaScript, Python, and SQLite. The system was tested with both legitimate and fraudulent URLs, analyzing parameters such as HTTPS usage, SSL certificate validity, and domain creation date. The results showed that the tool fulfills its purpose by adequately classifying websites into risk levels ('safe,' 'moderate,' and 'suspicious'), providing the user with a clear and immediate view of the reliability of the accessed pages. The integration between the modules proved to be stable and efficient. It is concluded that the proposed solution contributes to the democratization of digital security by providing an additional layer of protection and awareness while browsing.

KEYWORDS: *Digital security. Suspicious links. Browser extension. Flask. Online fraud.*

[1] Graduando do Curso de Engenharia da Computação da Universidade de Araraquara- UNIARA. Araraquara-SP.
[2] https://orcid.org/0000-0002-9341-0417 Docente do Curso de Sistemas de Informação da Universidade de Araraquara- UNIARA. Araraquara-SP. Bacharel em Direito pela Universidade de Araraquara (UNIARA), Mestre em Desenvolvimento Territorial e Meio Ambiente (UNIARA), Doutora em Alimentos e Nutrição (FCFAR/UNESP).
[3] https://orcid.org/0000-0003-2909-7611 Orientador. Docente Curso de Engenharia da Computação da Universidade de Araraquara- UNIARA. Araraquara-SP.

ISSN: 2675-6218 - RECIMA21

REVISTA CIENTÍFICA - RECIMA21 ISSN 2675-6218

RESUMEN
Este trabajo tiene como objetivo desarrollar una herramienta práctica y accesible orientada a la seguridad digital, capaz de ayudar a los usuarios en la verificación manual de enlaces sospechosos. La propuesta busca reducir la exposición a fraudes en línea, especialmente aquellos difundidos a través de redes sociales. Se realizaron investigaciones bibliográficas centradas en el levantamiento de datos sobre tácticas de fraude digital y prácticas de seguridad de la información. Posteriormente, se implementó una solución utilizando lenguajes y frameworks como JavaScript, Python y SQLite. El sistema fue probado con URL legítimas y fraudulentas, analizando parámetros como el uso de HTTPS, la validez de los certificados SSL y la fecha de creación del dominio. Los resultados demostraron que la herramienta cumple su función al clasificar adecuadamente los sitios en niveles de riesgo ("seguro", "moderado" y "sospechoso"), ofreciendo al usuario una visualización clara e inmediata de la confiabilidad de las páginas visitadas. La integración entre los módulos se mostró estable y eficiente. Se concluye que la solución propuesta contribuye a la democratización de la seguridad digital, al ofrecer una capa adicional de protección y concienciación durante la navegación.

PALABRAS CLAVE: *Seguridad digital. Enlaces sospechosos. Extensión de navegador. Flask. Fraude en línea.*

1. INTRODUÇÃO

A segurança digital dos usuários tem sido cada vez mais desafiada pela crescente sofisticação das fraudes *online*, particularmente aquelas propagadas por meio de plataformas de grande alcance, como Facebook e Instagram. Nesses ambientes, links para sites maliciosos, muitas vezes camuflados sob a aparência de ofertas atraentes ou informações relevantes, representam uma porta de entrada para golpes financeiros e roubo de dados pessoais.

A escolha deste tema se justifica por sua relevância social e pela oportunidade de aplicar conhecimentos técnicos no desenvolvimento de uma solução prática para um problema real. Autores como Schneier (2015), em suas obras sobre segurança da informação, enfatizam a importância da defesa em profundidade e da capacitação do usuário como elementos cruciais para a segurança digital. Nesse sentido, a ferramenta proposta visa complementar as medidas de segurança existentes, oferecendo ao usuário um controle ativo na avaliação da confiabilidade dos sites que acessa. Além disso, a natureza *client-side* da extensão, com possibilidade futura de expansão para um aplicativo móvel, busca minimizar a dependência de infraestruturas complexas e garantir a privacidade dos dados do usuário durante o processo de verificação, alinhando-se às melhores práticas de desenvolvimento seguro e às preocupações contemporâneas sobre a proteção da informação pessoal (Cavoukian, 2011).

A crescente sofisticação e a ampla disseminação de fraudes *online* representam uma ameaça constante à segurança e à confiança dos usuários na internet. Os golpes digitais têm apresentado um aumento alarmante, com prejuízos significativos tanto para indivíduos quanto para a economia como um todo. A facilidade com que links maliciosos são compartilhados em plataformas de mídia social, muitas vezes explorando a credulidade e a falta de conhecimento

técnico dos usuários, torna imperativa a criação de mecanismos de proteção acessíveis e eficazes.

Apesar do crescente número de recursos e informações sobre segurança *online* disponíveis aos usuários, observa-se uma persistente dificuldade na identificação de sites fraudulentos, especialmente aqueles habilmente disfarçados e disseminados por meio de plataformas de mídia social. A literatura e a prática demonstram que muitos usuários, particularmente os com menor conhecimento técnico, ainda são suscetíveis a golpes que exploram a confiança e a urgência de ofertas *online*. A ausência de uma ferramenta de verificação imediata e intuitiva, integrada ao fluxo de navegação e focada em indicadores acessíveis mesmo para leigos, contribui para a continuidade desse problema, resultando em perdas financeiras e comprometimento de dados pessoais.

A hipótese central é que o desenvolvimento e a disponibilização de uma extensão de navegador permitirá a verificação manual facilitada de sites suspeitos, por meio da análise de elementos como a correspondência de domínio, a validade do certificado SSL e a presença de padrões de conteúdo comumente associados a fraudes, pode aumentar significativamente a capacidade dos usuários de identificar e evitar golpes *online*. Acredita-se que uma ferramenta com interface simples e *feedback* claro sobre o nível de risco do site, acionável sob demanda no momento da suspeita, será mais eficaz na prevenção de fraudes do que a dependência exclusiva da conscientização geral ou de soluções automatizadas que nem sempre são transparentes ao usuário final.

Reconhecendo a vulnerabilidade dos usuários diante dessas táticas, este trabalho tem o objetivo de desenvolver de uma ferramenta intuitiva e de fácil acesso, inicialmente estruturada como uma extensão de navegador, a fim de capacitar o usuário na identificação de potenciais ameaças *online* antes que qualquer informação sensível seja comprometida.

O projeto centra-se na criação de um mecanismo de verificação manual, acionável pelo usuário ao se deparar com um site que levante suspeitas. Por meio da análise de diversos elementos da página *web*, a ferramenta buscará fornecer um panorama claro e objetivo sobre o nível de confiabilidade do site em questão. A meta principal é democratizar o acesso a informações de segurança, permitindo que mesmo usuários com menor conhecimento técnico realizem uma avaliação primária de riscos e, assim, fortaleçam sua postura defensiva no ambiente digital. A implementação inicial como uma extensão de navegador visa oferecer uma solução prática e integrada ao fluxo de navegação do usuário, com a possibilidade de futuras adaptações para outras plataformas, como aplicativos móveis.

O foco deste trabalho é capacitar os usuários a identificarem e evitarem fraudes *online*, por meio de uma ferramenta prática e acessível, concebida inicialmente como uma extensão de navegador para verificação manual de sites suspeitos, especialmente aqueles encontrados em

plataformas como Facebook e Instagram. O objetivo é alcançar uma extensão funcional que realize verificações essenciais em tempo real (como comparação de domínio, análise básica de SSL e identificação de padrões suspeitos), fornecendo *feedback* claro sobre o nível de risco, contribuindo para uma navegação mais segura e para a redução de golpes.

Foram realizadas pesquisas sobre táticas de fraude, desenvolvimento da extensão com as funcionalidades prioritárias, testes rigorosos para garantir usabilidade e eficácia, e a documentação completa do processo e resultados.

Foram realizadas pesquisas bibliográficas sobre fraudes *online*, técnicas utilizadas por golpistas e as abordagens existentes para detecção e prevenção. Dessas fraudes foram utilizados exemplos reais de sites fraudulentos e legítimos, a fim de identificar padrões e características que possam ser incorporados à lógica de verificação da extensão.

O desenvolvimento da extensão de navegador constituirá a principal etapa prática da pesquisa. Essa fase seguirá um ciclo iterativo e incremental, com as seguintes etapas: (1) planejamento e definição detalhada das funcionalidades; (2) desenvolvimento da estrutura básica da extensão; (3) implementação das funcionalidades de verificação (comparação de domínio, análise de SSL, análise de conteúdo); (4) testes unitários e de integração para garantir o correto funcionamento de cada componente e do sistema como um todo; (5) avaliação da usabilidade por meio de testes com usuários (se possível, mesmo que informais); e (6) refinamento e correção de falhas com base nos resultados dos testes. As ferramentas utilizadas incluirão um editor de código (VS Code), o navegador Google Chrome para desenvolvimento e testes da extensão, e, possivelmente, ferramentas *online* para análise de segurança de sites e consulta de informações de domínio.

2. REVISÃO BIBLIOGRÁFICA

2.1. Segurança da Informação no Contexto Digital

A segurança da informação é um campo que busca proteger os ativos digitais contra ameaças, assegurando que os dados estejam disponíveis apenas para usuários autorizados, que não sejam alterados indevidamente e que estejam acessíveis sempre que necessário. Esses três pilares, confidencialidade, integridade e disponibilidade (CID), são amplamente reconhecidos como fundamentais na área (Whitman; Mattord, 2018).

No contexto da navegação em sites, especialmente por meio de links recebidos via redes sociais, o risco de violação desses princípios aumenta significativamente. Sites falsos podem roubar informações pessoais, como senhas, dados bancários e documentos, comprometendo diretamente a confidencialidade e a integridade dos dados. A ausência de mecanismos visíveis de segurança, como o uso de HTTPS, pode deixar o usuário vulnerável, mesmo em ações simples como o preenchimento de um formulário.

O especialista Bruce Schneier (2015) argumenta que a segurança digital deve ser construída com base na defesa em profundidade, ou seja, a aplicação de múltiplas camadas de proteção para reduzir os riscos. Dentro dessa abordagem, destaca-se a importância de ferramentas que promovam a autonomia e a capacitação do usuário, especialmente frente ao crescimento das fraudes baseadas em engenharia social.

Nesse sentido, a extensão proposta neste trabalho atua como uma camada adicional de segurança, oferecendo uma forma prática de identificar indícios de fraude a partir de verificações que não exigem conhecimento técnico, mas que são baseadas em princípios sólidos de segurança da informação.

2.2. Fraudes *Online* e Engenharia Social

2.2.1. Táticas Comuns de Golpistas

Fraudes *online* são, em sua essência, formas de engenharia social aplicadas a ambientes digitais. Os criminosos utilizam técnicas como *phishing, spoofing, malvertising* (publicidade maliciosa) e clonagem de sites para enganar usuários e obter informações sigilosas ou induzi-los a ações prejudiciais, como transferências financeiras.

O *phishing* é uma das práticas mais comuns e consiste na criação de sites que imitam visualmente páginas legítimas, com a intenção de capturar dados pessoais. O *spoofing,* por sua vez, está relacionado à falsificação de elementos como nomes de domínio, endereços de *e-mail* ou URLs, tornando ainda mais difícil a identificação da fraude (Oliveira *et al.,* 2021).

De acordo com o Relatório da Kaspersky (2022), ataques de *phishing* representaram mais de 35% das tentativas de fraude digital no Brasil, colocando o país entre os principais alvos dessas ações na América Latina. Tais dados evidenciam a necessidade de ferramentas que permitam verificar, de forma rápida e direta, a autenticidade dos sites acessados pelos usuários.

2.2.2. O Papel das Redes Sociais na Disseminação de Golpes

As redes sociais são ambientes com alta taxa de compartilhamento de conteúdo e baixo controle de veracidade, o que as torna ideais para a disseminação de links maliciosos. Plataformas como Facebook, Instagram e WhatsApp são frequentemente utilizadas para veicular ofertas falsas, sorteios inexistentes e anúncios clonados, com o objetivo de levar o usuário a sites fraudulentos.

Segundo dados da Federação Brasileira de Bancos (FEBRABAN, 2023), mais de 70% dos golpes relatados no último ano envolviam interações em redes sociais, sendo que muitos dos usuários sequer perceberam que estavam sendo redirecionados para sites falsos até sofrerem prejuízos.

Essa realidade reforça a urgência de criar soluções acessíveis, integradas ao fluxo de navegação e adaptadas à experiência do usuário leigo, que muitas vezes não conhece os sinais técnicos de alerta presentes em sites falsificados.

2.3. Ferramentas e Tecnologias Aplicadas

2.3.1. Desenvolvimento de Extensões de Navegador

O desenvolvimento de extensões para navegadores como o Google Chrome é viabilizado por meio de APIs fornecidas pelo próprio navegador, permitindo a criação de funcionalidades personalizadas que interagem com páginas *web*. A estrutura básica de uma extensão inclui o arquivo *manifest.json*, que define permissões, *scripts, interface (popup)* e ações de fundo.

2.3.2. Análise de Certificados SSL e Domínios

A verificação da presença de um certificado SSL *(Secure Sockets Layer)* é uma das práticas básicas de segurança digital. O SSL permite a criptografia da comunicação entre o navegador e o servidor, impedindo a interceptação de dados. Sites que utilizam SSL são identificados pelo prefixo "https://" e pelo ícone de cadeado exibido no navegador.

Apesar de sua importância, a presença de SSL não garante a legitimidade de um site, visto que certificados gratuitos são facilmente obtidos por golpistas. Ainda assim, sua ausência é um indicativo de risco e será tratada como um alerta relevante pela extensão.

Outro critério importante é a análise de domínios suspeitos. Muitos golpes utilizam domínios semelhantes aos de empresas conhecidas (por exemplo, "mercado1ivre.com" em vez de "mercadolivre.com.br"), prática conhecida como *typosquatting.* A verificação da URL, bem como o uso de ferramentas de consulta como Whois e Google Safe Browsing, pode aumentar significativamente a assertividade da ferramenta proposta.

2.3.3. Interface de Usuário e Usabilidade

Para que a ferramenta atinja seu público-alvo, sua interface precisa ser intuitiva, clara e funcional. Conforme Nielsen (2000), um bom sistema é aquele que permite ao usuário concluir tarefas com eficiência, mesmo sem treinamento prévio.

A extensão terá como princípio o acionamento manual, permitindo ao usuário verificar a confiabilidade de um site sempre que suspeitar de sua legitimidade. A interface mostrará alertas visuais simples, como indicadores de "site confiável", "atenção" ou "possivelmente fraudulento", com base em critérios objetivos.

O uso de ícones, cores e mensagens diretas facilitará a tomada de decisão por parte do usuário, incentivando a navegação segura sem a necessidade de conhecimento técnico aprofundado.

REVISTA CIENTÍFICA - RECIMA21 ISSN 2675-6218

3. DESENVOLVIMENTO DE UMA VERIFICAÇÃO MANUAL DE SITES FRAUDULENTOS

Esta seção, apresenta o desenvolvimento da arquitetura do sistema, destacando a integração entre os módulos que compõem a solução proposta. O objetivo é demonstrar como cada componente contribui para a execução das verificações e para o retorno das informações ao usuário.

O sistema consiste em uma solução *web* distribuída composta por três camadas principais: (1) uma extensão de navegador, (2) um *backend* implementado em *Flask* e (3) uma interface de *dashboard* acessível via navegador. A extensão é responsável por capturar a URL da aba ativa e enviá-la para análise. O *backend* realiza a verificação da segurança do site com base em critérios definidos e armazena os resultados em um banco de dados SQLite. Posteriormente, os dados são visualizados por meio de uma *dashboard web* que apresenta o histórico de sites verificados com suas respectivas classificações de risco.

3.1. Implementação da Extensão de Navegador

Esta subseção, aborda o processo de criação da extensão responsável pela interação direta com o usuário. Aqui são explicadas as principais configurações do projeto, os arquivos envolvidos e a forma como ocorre a comunicação com o servidor.

A extensão foi desenvolvida para o navegador Google Chrome, utilizando a API de extensão em sua versão Manifest V3. O arquivo *manifest.json* define as permissões e configurações principais, incluindo a execução de um *script* responsável por capturar a URL da aba ativa e acionar uma requisição *POST* para o servidor *Flask.*

A interface da extensão é composta por um *popup* que apresenta um botão de verificação e uma área de exibição de resultados. O *popup.js* gerencia a interação com o usuário, incluindo o envio da URL e a exibição de mensagens visuais de risco (como "Site seguro", "Atenção" ou "Site suspeito") com cores indicativas.

A Figura 1 ilustra um pequeno trecho do código que realiza a função central da extensão: capturar a URL da aba atual e enviá-la para análise pelo *backend Flask*.

Figura 1. Captura da URL ativa e envio para *backend*

```
chrome.tabs.query({ active: true, currentWindow: true }, function (tabs) {
  let url = tabs[0].url;
  fetch("http://127.0.0.1:5000/verificar", {
    method: "POST",
    headers: {"Content-Type": "application/json"},
    body: JSON.stringify({ url: url })
  })
```

Fonte: O autor (2025)

ISSN: 2675-6218 - RECIMA21

REVISTA CIENTÍFICA - RECIMA21 ISSN 2675-6218

3.1.1. Implementação do *Backend Flask*

Nesta subseção, o texto especifica o funcionamento da API, que realiza as verificações técnicas e classifica os sites conforme os critérios de segurança estabelecidos. O objetivo é mostrar como os diferentes elementos de verificação são aplicados e combinados para gerar um resultado final confiável.

O *backend* foi desenvolvido com o *framework Flask* em Python. A API principal recebe URLs via método *POST* na rota /verificação. Ao receber uma URL, o servidor executa uma série de análises:

- Verifica se a URL utiliza HTTPS;
- Verifica a validade real do certificado SSL (usando a biblioteca ssl);
- Acessa o conteúdo HTML da página e procura palavras-chave suspeitas;
- Consulta a data de criação do domínio via WHOIS;
- Gera uma pontuação de risco com base na soma de fatores;
- Classifica o site como "seguro", "moderado" ou "fraudulento".

Esse trecho da Figura 2 inicia o dicionário de resultado e inicia a lógica de verificação dos critérios de segurança da URL recebida.

Figura 2 . Trecho da função principal de análise de site

```
def verificar_site(url):
    resultado = {
        "url": url,
        "tem_https": url.startswith("https://"),
        "status_code": None,
        "dominio_recente": False,
        "dias_desde_criacao": None,
        "palavras_suspeitas": False,
        "certificado_ssl_valido": False,
        "possivelmente_fraudulento": False,
        "score_risco": 0,
        "motivos_risco": []
    }
```

Fonte: O autor (2025)

ISSN: 2675-6218 - RECIMA21

REVISTA CIENTÍFICA - RECIMA21 ISSN 2675-6218

A Figura 3 apresenta um trecho que realiza a conexão segura com o servidor para extrair o certificado SSL e validar sua autenticidade e validade temporal.

Figura 3. Verificação real de validade do certificado SSL

```
with socket.create_connection((hostname, 443), timeout=3) as sock:
    with context.wrap_socket(sock, server_hostname=hostname) as ssock:
        cert = ssock.getpeercert()
```

Fonte: O autor (2025)

3.1.2. Banco de Dados e Armazenamento

Esta seção descreve o armazenamento das informações obtidas durante as verificações, ressaltando a importância do banco de dados para a persistência e consulta dos resultados. Também são apresentados os principais campos e sua função dentro do sistema.

Foi utilizado um banco de dados SQLite local, com inicialização automática via *script* (inicializar banco.py). A tabela principal armazena:

- ID;
- URL verificada;
- Data e hora da verificação;
- Score de risco (número de fatores de alerta);
- *Status* classificado (seguro, moderado, suspeito).

Essa estrutura da Figura 4 permite consultas simples e eficientes para alimentar a *dashboard*.

ISSN: 2675-6218 - RECIMA21

Figura 4. Criação da tabela SQLite

```
cursor.execute("""
CREATE TABLE IF NOT EXISTS sites (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    url TEXT NOT NULL,
    status_code TEXT,
    tem_https BOOLEAN,
    certificado_ssl_valido BOOLEAN,
    dominio_recente BOOLEAN,
    palavras_suspeitas BOOLEAN,
    score_risco INTEGER,
    possivelmente_fraudulento BOOLEAN,
    motivos_risco TEXT,
    data_analisada TIMESTAMP DEFAULT CURRENT_TIMESTAMP
)
""")
```

Fonte: O autor (2025)

3.2. Fluxo de Funcionamento da Verificação

Nesta subseção, é apresentado o funcionamento completo do sistema, desde o momento em que o usuário realiza a verificação até o retorno dos resultados. O objetivo é ilustrar o caminho percorrido pelos dados e a integração entre as camadas do projeto.

1. O usuário acessa um site no navegador;
2. Ao suspeitar de algo, clicar no ícone da extensão e aciona o botão de verificação;
3. A extensão envia a URL atual para o *backend;*
4. O *backend* analisa os critérios de segurança e classifica o site;
5. O resultado é retornado à extensão e exibido ao usuário em tempo real;
6. Os dados da verificação ficam salvos para posterior consulta na *dashboard.*

3.2.1. Registro de Execução e Comunicação entre os Componentes

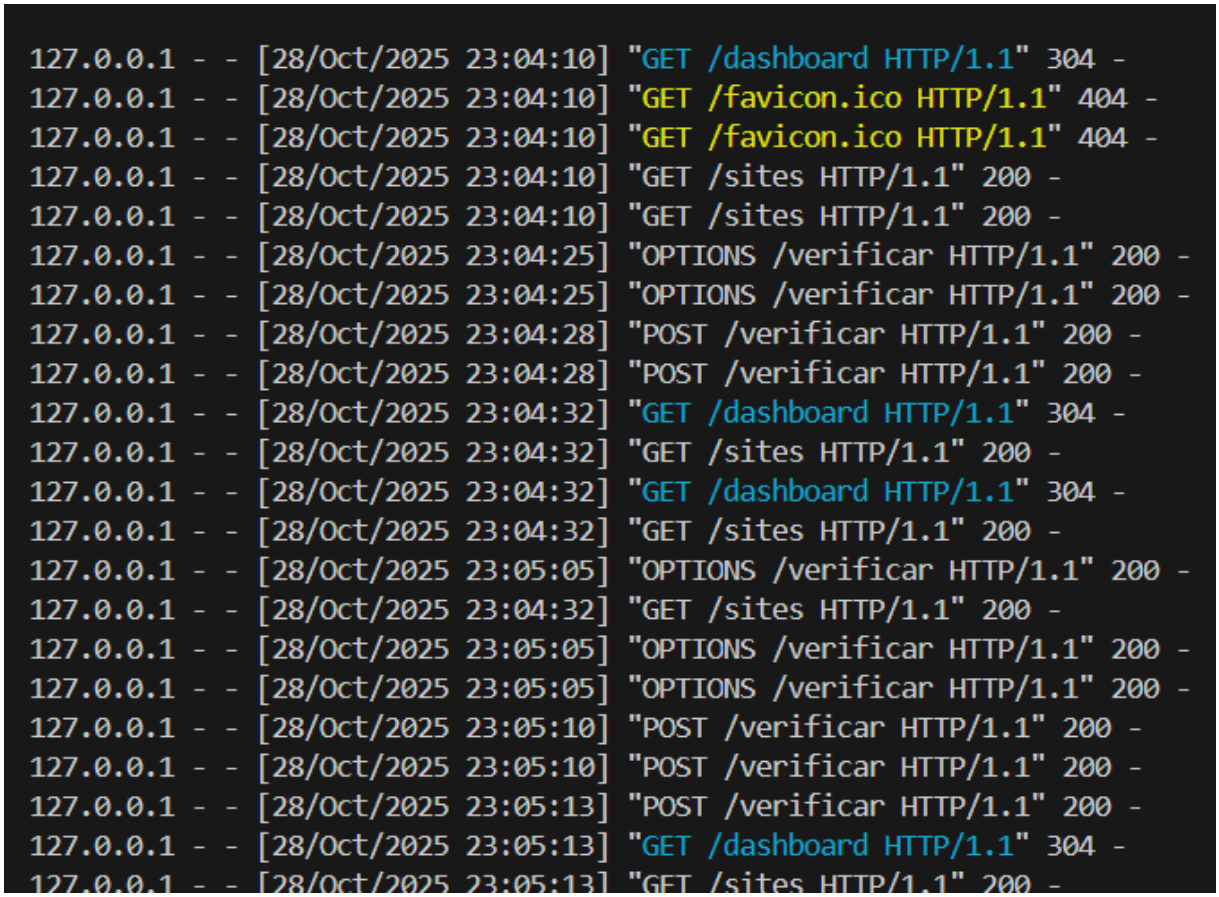
Com o objetivo de validar o funcionamento das rotas e a troca de informações entre as diferentes camadas do sistema, foi realizado um acompanhamento detalhado do comportamento do servidor *Flask* durante as operações de verificação. Esta etapa serve para demonstrar, de forma prática, que a extensão, o *backend* e a *dashboard* interagem corretamente, garantindo que as requisições e respostas ocorram conforme o planejado no fluxo de funcionamento descrito anteriormente.

Durante os testes de integração entre a extensão, o *backend Flask* e a *dashboard*, foi possível observar em tempo real o fluxo de comunicação entre as camadas do sistema por meio do log gerado no terminal. Esse registro mostra as requisições do tipo *GET*, *POST* e *OPTIONS*,

REVISTA CIENTÍFICA - RECIMA21 ISSN 2675-6218

realizadas pela extensão e pela interface *web*, evidenciando o correto funcionamento do servidor e o recebimento das URLs enviadas para análise. Cada linha indica o momento em que a extensão solicita a verificação de um site, o *backend* processa a requisição e retorna a resposta classificada, confirmando a execução contínua e estável das rotas implementadas.

Figura 5. Log de execução do servidor *Flask* durante as requisições de verificação



Fonte: O autor (2025)

3.3. *Dashboard* de Monitoramento

Esta etapa descreve a interface visual responsável por exibir os resultados das verificações. O texto apresenta sua estrutura, usabilidade e papel dentro da proposta do sistema.

A *dashboard* é uma página HTML *(dashboard.html)* que consome os dados armazenados no banco por meio de uma rota no *backend.* A página apresenta uma tabela com as URLs analisadas, data da verificação e nível de risco. Cores são utilizadas para facilitar a identificação visual dos status (verde para seguro, amarelo para moderado, vermelho para suspeito). Além da dashboard, os resultados também podem ser visualizados em tempo real no *popup* da extensão,

acionado pelo usuário no momento da verificação. Como exemplo, a Figura 6 apresenta um site classificado como suspeito, sem uso de HTTPS válido, enquanto a Figura 7 ilustra um site seguro, com certificado SSL ativo, e a Figura 8 mostra a dashboard exibindo os sites verificados.

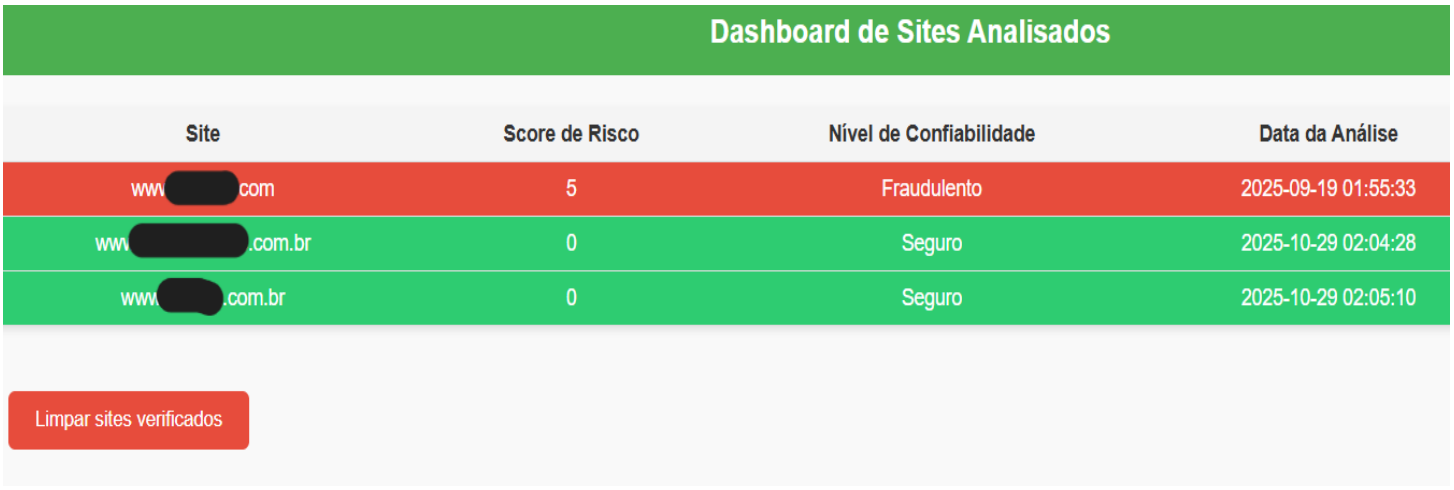
Figura 6. Site suspeito

Fonte: O autor (2025)

Figura 7. Site seguro

Fonte: O autor (2025)

Figura 8. *Dashboard*



Fonte: O autor (2025)

4. CONSIDERAÇÕES

Este trabalho teve como objetivo desenvolver uma solução prática e acessível para auxiliar usuários na identificação de sites potencialmente fraudulentos, especialmente aqueles disseminados por redes sociais. A ferramenta proposta: uma extensão de navegador com *backend* em *Flask* e interface de *dashboard*, demonstrou-se funcional e eficaz na verificação manual de links suspeitos, aplicando critérios técnicos simples, mas relevantes, como análise de HTTPS, validade de certificados SSL, data de criação de domínio e presença de padrões de conteúdo suspeitos.

Durante o desenvolvimento, observou-se que a integração entre a extensão e o servidor *Flask* proporcionou uma comunicação eficiente, permitindo a análise em tempo real das URLs visitadas. O uso do banco de dados SQLite mostrou-se adequado para armazenar e consultar o histórico de verificações, contribuindo para a rastreabilidade e auditoria dos resultados. A *dashboard* complementa o sistema ao oferecer uma visualização organizada e intuitiva, possibilitando ao usuário acompanhar o histórico e compreender de forma imediata o nível de segurança de cada site analisado.

Os resultados demonstram que o projeto cumpre seu propósito central de democratizar o acesso à segurança digital, oferecendo uma camada adicional de proteção que independe de conhecimento técnico avançado. A simplicidade da interface e a clareza dos alertas permitem que qualquer usuário possa realizar verificações rápidas e tomar decisões mais seguras ao navegar pela internet.

Como trabalhos futuros, propõe-se a expansão da solução para dispositivos móveis, permitindo integração com navegadores de *smartphones*, e a inclusão de métodos automatizados de detecção, como análise de reputação de domínios via APIs de segurança e modelos de aprendizado de máquina. Também é possível integrar a ferramenta com sistemas de denúncia e *feedback* colaborativo, fortalecendo sua base de dados e aumentando sua precisão.

Reforça-se que, este projeto representa uma contribuição significativa para o campo da segurança digital acessível, combinando tecnologia, usabilidade e conscientização do usuário. A extensão desenvolvida serve como um passo concreto na direção de uma internet mais segura, consciente e inclusiva.

REFERÊNCIAS

CALLEGATI, F.; CAMPOS, M.; PAGANO, M. *Security issues in the HTTPS protocol*. **IEEE Communications & Security**, v. 7, p. 30-36, 2009.

CAVOUKIAN, A. Privacy by Design: The 7 Foundational Principles. **Ontario Information and Privacy Commissioner**, 2011.

FEBRABAN – FEDERAÇÃO BRASILEIRA DE BANCOS. **Relatório Anual de Segurança Digital**. São Paulo: FEBRABAN, 2023.

GOOGLE. Extension Manifest V3. Chrome Developers. Disponível em: https://developer.chrome.com/docs/extensions/mv3. Acesso em: 05 abr. 2025.

KASPERSKY. Relatório de Cibersegurança no Brasil. **Kaspersky Security Bulletin**, 2022. Disponível em: https://www.kaspersky.com.br. Acesso em: 28 mar. 2025.

NIELSEN, J. **Designing Web Usability:** The Practice of Simplicity. Hoboken, NJ: New Riders Publishing, 2000.

OLIVEIRA, A. P.; SILVA, M. C.; GOMES, R. M. Cibercrimes e Engenharia Social: Um Estudo sobre Práticas Criminosas na Internet. **Revista Brasileira de Segurança da Informação**, v. 8, n. 1, p. 42-56, 2021.

SCHNEIER, B. **Data and Goliath**: The Hidden Battles to Collect Your Data and Control Your World. W. W. New York: Norton & Company, 2015.

WHITMAN, M. E.; MATTORD, H. J. **Principles of Information Security**. 6. ed. Boston: Cengage Learning, 2018.