INSTITUTO FEDERAL DE RONDÔNIA
CAMPUS PORTO VELHO CALAMA
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

ALBERT JOSSUE OJEDA GOMEZ

PHISHIELD: FERRAMENTA WEB PARA DETECÇÃO DE URLS MALICIOSAS

Porto Velho/RO
2025

ALBERT JOSSUE OJEDA GOMEZ

PHISHIELD: FERRAMENTA WEB PARA DETECÇÃO DE URLS MALICIOSAS

Artigo entregue como Trabalho de Conclusão de Curso ao Instituto Federal de Educação, Ciência e Tecnologia de Rondônia (IFRO), *Campus* Porto Velho Calama, como requisito parcial para obtenção do grau de Tecnólogo, junto ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Leandro Ferrarezi Valiante

Porto Velho/RO
2025

Ficha catalográfica elaborada pelo Sistema Gerador de Ficha Catalográfica do IFRO.

Gomez, Albert Jossue Ojeda.
Phishield: ferramenta web para detecção de URLs maliciosas / Albert Jossue Ojeda Gomez. - Porto Velho, 2025.
19 f. : il.

Orientador(a): Prof. Me. Leandro Ferrarezi Valiante.

Trabalho de Conclusão de Curso (Superior de Tecnologia em Análise e Desenvolvimento de Sistemas) – Instituto Federal de Educação, Ciência e Tecnologia de Rondônia - IFRO, Porto Velho, 2025.

1. Phishing. 2. Extensão web. 3. URLs. I. Valiante, Leandro Ferrarezi (orient.). II. Instituto Federal de Educação, Ciência e Tecnologia de Rondônia - IFRO. III. Título.

Bibliotecário(a) Responsável: Miria Santana Veiga, CRB-11/898

ALBERT JOSSUE OJEDA GOMEZ

PHISHIELD: FERRAMENTA WEB PARA DETECÇÃO DE URLS MALICIOSAS

A banca examinadora, abaixo listada, aprova o Trabalho de Conclusão de Curso “PHISHIELD: FERRAMENTA WEB PARA DETECÇÃO DE URLs MALICIOSAS” elaborado por “Albert Jossue Ojeda Gomez” como requisito parcial para obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas, pelo Instituto Federal de Educação, Ciência e Tecnologia de Rondônia.

Porto Velho/RO, 27/11/2025

Comissão Examinadora

LEANDRO FERRAREZI VALIANTE:82694095291

Assinado de forma digital por LEANDRO FERRAREZI VALIANTE:82694095291
Dados: 2025.12.01 11:18:34 -04'00'

Prof. Leandro Ferrarezi Valiante - IFRO
(Orientador)

gov.br Documento assinado digitalmente
CELSO GUEDES GOMES
Data: 01/12/2025 16:35:22-0300
Verifique em https://validar.iti.gov.br

Prof. Celso Guedes Gomes - IFRO

gov.br Documento assinado digitalmente
CLEYTON PEREIRA DOS SANTOS
Data: 04/12/2025 11:37:53-0300
Verifique em https://validar.iti.gov.br

Prof. Cleyton Pereira dos Santos - IFRO

PHISHIELD: FERRAMENTA WEB PARA DETECÇÃO DE URLS MALICIOSAS

RESUMO: As ameaças cibernéticas são recorrentes no cenário da segurança digital, afetando e colocando em risco milhões de usuários na internet. O phishing é uma dessas ameaças ainda presentes no cotidiano; essa técnica, que utiliza links disfarçados como legítimos, tem como objetivo roubar dados sensíveis e informações pessoais. Embora existam ferramentas automatizadas de detecção, muitas delas não são acessíveis ao público em geral, seja pelo custo elevado ou pela complexidade de uso. Estudos indicam que características presentes na estrutura das URLs, como tamanho, formatação e uso de termos suspeitos, podem servir como indícios úteis na identificação de golpes online. Diante disso, este estudo propõe o PhiShield, uma extensão web desenvolvida para avaliar a segurança dos links suspeitos. A proposta busca oferecer uma solução simples, gratuita e acessível, capaz de alertar rapidamente os usuários sobre potenciais ameaças, contribuindo para tornar a segurança digital mais acessível a todos.

PALAVRAS-CHAVE: Phishing; Extensão web; URLs.

ABSTRACT: Cyber threats are frequent in the digital security landscape, affecting and endangering millions of internet users. Phishing is one of the threats still commonly observed; this technique, which uses links disguised as legitimate, aims to steal sensitive data and personal information. Although automated detection tools exist, many are not accessible to the general public due to high cost or complexity of use. Studies indicate that features in URL structures, such as size, formatting, and the presence of suspicious terms, can serve as useful clues for identifying online scams. Therefore, this study proposes PhiShield, a web extension developed to assess the security of suspicious links. The proposal aims to offer a simple, free, and accessible solution capable of quickly alerting users to potential threats, contributing to making digital security more accessible to everyone.

KEYWORDS: Phishing Detection; Web Extension; Cybersecurity.

1 INTRODUÇÃO

O grande volume de pessoas acessando e compartilhando dados na internet tem se tornado um alvo recorrente para os cibercriminosos, o que exige atenção dos usuários com relação ao consumo de informações digitais, já que nem todo conteúdo disponível na Internet é confiável. Acreditar em tudo o que se encontra na web facilita que agentes maliciosos explorem a confiança dos usuários com táticas enganosas e criativas que induzem o compartilhamento de informações pessoais.

Dentre as principais ameaças destaca-se o **phishing**, um termo cujo significado é pescar, e que começou a ser usado em 1995. Este termo refere-se ao ato de enganar os usuários com links suspeitos ou anexos infectados com o objetivo de roubar dados sensíveis, senhas, números de cartões e credenciais de acesso (Phishing.org, 2021).

Alguns testes simulados pela Proofpoint (2023) mostraram que 9.3% dos usuários clicaram em links de e-mails maliciosos, evidenciando como uma das principais fraudes digitais da internet (links falsos) representa significativo risco.

O aumento no número de páginas falsas no cenário digital brasileiro tem causado crescente preocupação. O CERT.br (2025) aponta que, entre janeiro e agosto de 2025, foram identificadas 4.629 páginas falsas, das quais 3.764 afetaram organizações brasileiras. Entre os setores mais visados para golpes baseados em links fraudulentos está o setor financeiro, com mais de 1.588 páginas falsas registradas. Além disso, foram identificados mais de 2.499 endereços de internet e 233 redes de computadores usados para hospedar essas páginas, o que demonstra como grupos mal-intencionados utilizam esquemas complexos e ferramentas sofisticadas para atrair suas vítimas.

A infraestrutura massiva, que exige coordenação e recursos significativos, comprova que o uso de links falsos para aplicar golpes via Internet não é apenas uma prática isolada de cibercriminosos individuais, mas sim uma atividade organizada e altamente lucrativa. Esses dados reforçam o impacto causado por tais ataques, já que não afetam apenas usuários comuns, mas também empresas de grande porte e serviços críticos.

Entre as diversas modalidades de ataques cibernéticos, o *phishing* destaca-se pela frequência e eficácia em explorar vulnerabilidades humanas e tecnológicas. A escalada dessa ameaça cibernéticas tem se tornado cada vez mais evidente no panorama global.

De acordo com o Relatório Anual de Fatores Humanos da Proofpoint (2022), os cibercriminosos continuam a explorar vulnerabilidades humanas e tecnológicas com crescente criatividade e sofisticação. O estudo, que analisou mais de 49 bilhões de URLs (sigla para *Uniform Resource Locator*, ou Localizador Padrão de Recursos) e

$2,6$ bilhões de mensagens de e-mail por dia ao longo de 2021, evidencia a frequência e a escala alarmante de tentativas de ataques como *phishing*. Esse cenário global se reflete com a intensidade no Brasil, onde, apenas no ano passado, foram registradas mais de 100 bilhões de tentativas de ataques cibernéticos, conforme o levantamento da Fortinet citado pela Algar(2024) (Telecom, 2024).

Considerando essa vasta quantidade de informações de ameaças, que tem o potencial de desequilibrar operações e comprometem a integridade de dados, a adoção de medidas robustas de segurança tornou-se um imperativo para organizações de todos os portes.

Com isso, faz-se necessário agir em busca de soluções que ajudem a diminuir os riscos associados à prática de divulgação de links falsos. Nesse contexto, este estudo apresenta o desenvolvimento de uma extensão web de código aberto voltada a detecção de links de sites maliciosos, buscando oferecer acessibilidade e fácil utilização para usuários. A proposta se apoia na necessidade crescente de ferramentas de apoio à segurança digital, ressaltando a importância de integrar tecnologia e orientação do usuário para fortalecer a capacidade de identificar e evitar ameaças cibernéticas cada vez mais sofisticadas

1.1 Objetivos

1.1.1 Objetivo geral

Desenvolver uma extensão web de código aberto capaz de identificar URLs potencialmente perigosos ou fraudulentos, está extensão visa mitigar o risco de ataques nos quais cibercriminosos tentam enganar usuários para que revelem informações confidenciais. Além disso a ferramenta contribuirá para a proteção dos usuários ao detectar automaticamente anomalias e padrões suspeitos em URLs que possam indicar tentativas de phishing.

1.1.2 **Objetivos específicos**

(i) Analisar as principais técnicas utilizadas em ataques de phishing e como elas exploram vulnerabilidades humanas e tecnológicas;

(ii) Projetar e implementar uma extensão web para detecção de URLs suspeitas;

(iii) Desenvolver uma interface intuitiva que apresente ao usuário informações claras sobre o nível de risco de uma página;

(iv) Realizar testes de detecção e desempenho da ferramenta, avaliando sua eficácia na identificação de sites suspeitos;

(v) Integrar mecanismos de feedback na extensão, explicando os critérios de detecção que fundamentam a classificação de risco.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Segurança da informação

De acordo com (Lyra, 2019), a informação vai além de um simples dado isolado, representando um conjunto de dados organizados que adquirem significado dentro de um contexto específico. Nas organizações, a informação sempre foi um ativo crítico, essencial para a tomada de decisões estratégicas, aumento da competitividade e eficiência operacional.

Diante do valor crítico da informação, a necessidade de protegê-la tornou-se fundamental. Conforme sintetizado por (Lyra, 2019), a partir de normas e autores consagrados, a segurança da informação pode ser definida como a proteção dos ativos informacionais a fim de preservar três pilares fundamentais:

- **Confidencialidade**: Garantia de que a informação é acessível apenas por usuários autorizados
- **Integridade**: Garantia de que a informação permanece inalterada
- **Disponibilidade**: Garantia de que a informação está acessível quando necessária.

Além desses pilares centrais, o autor também destaca conceitos complementares que reforçam a segurança da informação, tais como: autenticidade, que corresponde à confirmação da identidade do usuário; não repúdio, que consiste na impossibilidade de negar a autoria de uma ação; e auditoria, que é a capacidade de registrar e rastrear atividades dentro do sistema (Lyra, 2019).

No escopo desta pesquisa, os princípios apresentados são diretamente impactados por ameaças como o *phishing*, cujo objetivo é violar a confidencialidade e a integridade de dados por meio da engenharia social e da imitação de páginas legítimas. Assim, compreender os conceitos fundamentais da segurança da informação é essencial para embasar o desenvolvimento de soluções como o “PhiShield”, que busca contribuir para a proteção dos usuários contra esse tipo de ataque.

A implementação da extensão de navegador “PhiShield” foi concebida com base em um *stack* tecnológico selecionado criteriosamente para atender aos requisitos específicos de uma ferramenta de segurança web, priorizando a eficiência na

análise de URLs, a segurança no processamento de dados e a compatibilidade total com o ecossistema de extensões do navegador Google Chrome. Nesta seção, são detalhados os componentes tecnológicos fundamentais e suas respectivas funções no contexto do projeto.

2.2 Tecnologias de Desenvolvimento para Web

O HTML (*HyperText Markup Language*, ou Linguagem de Marcação de Hipertexto) é uma linguagem de marcação padrão para a criação de documentos web, conforme definido pelas especificações do *World Wide Web Consortium* (MDN Web Docs, 2024). Trata-se de uma linguagem baseada em elementos demarcados por *tags* que definem a estrutura semântica e hierárquica do conteúdo, organizando-o em seções, parágrafos, cabeçalhos, listas e outros componentes. A evolução para o HTML5 introduziu elementos semânticos como `<header>`, `<nav>`, `<section>`, `<article>` e `<footer>`, que permitem uma representação mais precisa da estrutura do documento.

O CSS (*Cascading Style Sheets*, ou Folhas de Estilo em Cascata) é uma linguagem de folhas de estilo desenvolvida pelo World Wide Web Consortium (W3C) para controlar a apresentação visual de documentos estruturados (World Wide Web Consortium, 2024). O seu papel principal é dissociar a estrutura **(HTML)** da sua aparência visual, permitindo o controle preciso de *layout*, cores, tipografia e efeitos de transição.

O modelo de formatação do CSS baseia-se em três conceitos fundamentais: o *box-model*, que define como os elementos são renderizados como caixas retangulares; o *cascade*, que determina a prioridade entre regras conflitantes; e a herança, que permite que propriedades sejam passadas de elementos pais para filhos. A especificação CSS3 introduziu recursos avançados como *flexbox*, *grid layout*, *media queries* para responsividade, variáveis CSS e animações, ampliando significativamente as capacidades de estilização.

O JavaScript é classificado pela MDN Web Docs (2025a) como uma linguagem de programação de propósito geral, capaz de operar tanto por interpretação quanto por compilação *just-in-time* (JIT). A linguagem se destaca por oferecer funções de primeira classe e por adotar um modelo de tipagem dinâmica, características que contribuem para sua flexibilidade no desenvolvimento web.

Conforme detalhado na documentação do MDN Web Docs (2025a), o JavaScript é baseado em protótipos, realiza gerenciamento automático de memória por meio do mecanismo de *garbage collection* e fornece suporte a múltiplos paradigmas de programação, incluindo os estilos orientado a objetos, funcional e imperativo. Essas propriedades tornam a linguagem especialmente adequada para aplicações que exigem interação dinâmica, manipulação de eventos e execução de lógica diretamente no na-

vegador.

O Python é uma linguagem de programação de alto nível, orientada a objetos de tipagem dinâmica e forte, interpretada e sobretudo interativa. Conforme Borges (2014), a linguagem apresenta uma sintaxe básica clara e concisa, o que leva à legibilidade do código fonte e, consequentemente, torna a linguagem altamente produtiva.

A linguagem Python é "multiparadigma", suportando programação modular, funcional e orientada a objetos, onde até mesmo os tipos básicos são tratados como objetos. De acordo com a Python Software Foundation (2025), a linguagem inclui diversas estruturas como listas, dicionários, tipos para manipulação de data/hora e números complexos. É distribuída como software de código aberto sob licença compatível com a GPL, porém menos restritiva, permitindo sua incorporação em produtos proprietários. A especificação e evolução da linguagem são mantidas pela *Python Software Foundation*.

Flask é um *microframework* WSGI leve implementado para aplicações web desenvolvidas em Python. O termo WSGI (*Web Server Gateway Interface*, ou Interface de *Gateway* para Servidores Web) refere-se a um padrão que define como servidores web e aplicações Python devem se comunicar, garantindo interoperabilidade entre diferentes implementações. Conforme a documentação oficial do Pallets Projects (2025), o Flask fornece uma base minimalista e de baixo acoplamento no desenvolvimento web, mantendo a capacidade de expansão para aplicações complexas.

A arquitetura do Flask oferece os componentes essenciais para aplicações web, fornecendo os componentes básicos para o desenvolvimento web. Sua natureza modular lhe permite a incorporação seletiva de funcionalidades adicionais por meio de um robusto sistema de extensões. O *framework* é tecnicamente fundamentado no *toolkit* "WSGI Werkzeug" e no mecanismo de *template* "Jinja", garantindo compatibilidade com padrões web e eficiência na renderização.

A *JavaScript Object Notation* (JSON), em português Notação de Objetos *JavaScript*, é um formato textual amplamente utilizado para representar e transmitir dados estruturados em aplicações web. Sua sintaxe deriva da estrutura de objetos do *JavaScript*, porém o formato é independente de linguagem, ou seja, pode ser lido, interpretado e gerado por diferentes ambientes de programação. Em termos práticos, o JSON organiza informações em pares chave–valor e permite a criação de hierarquias de dados utilizando tipos simples como números, cadeias de caracteres, valores booleanos, listas e objetos. Por ser leve, legível e facilmente convertido entre texto e estruturas internas de dados, o JSON tornou-se o padrão predominante para troca de informações entre cliente e servidor, sendo utilizado no **PhiShield** para registrar e apresentar os resultados completos da análise de URLs (MDN Web Docs, 2025b).

A *Application Programming Interface* (API), ou Interface de Programação de Aplicações, consiste em um conjunto estruturado de regras, padrões e protocolos que permite a comunicação entre diferentes sistemas de software. Seu papel central é definir como aplicações podem solicitar dados, executar operações e acessar funcionalidades de maneira consistente e bem documentada. Conforme descrito pela documentação oficial da Oracle, uma API atua como um intermediário responsável por receber solicitações, processar as ações necessárias e retornar respostas em formatos padronizados, geralmente utilizando estruturas como JSON ou XML (Oracle, 2025).

3 METODOLOGIA

O desenvolvimento do **PhiShield** começou com uma revisão da literatura especializada, com o objetivo de mapear as soluções existentes no enfrentamento ao *phishing*, e avaliar a viabilidade técnica da proposta. Posteriormente Foi conduzida uma análise focada no desenvolvimento de extensões para navegadores, considerando as funcionalidades disponibilizadas pelas APIs nativas dessas plataformas e suas respectivas regras de segurança. Essa etapa permitiu compreender quais permissões, recursos e limitações seriam aplicáveis ao projeto da ferramenta.

Posteriormente foi conduzida uma análise detalhada das técnicas contemporâneas de *phishing*, com foco específico em URLs fraudulentas e métodos de camuflagem. Com base nesse estudo, procedeu-se ao planejamento da arquitetura do sistema, definindo os componentes necessários e suas interações.

A implementação adotou um processo continuo e incremental,iniciando pela construção das funções essenciais de detecção e progredindo para métodos mais sofisticados. Foram desenvolvidos algoritmos específicos para analisar a estrutura da URLS, focando na identificação de padrões suspeitos e na detecção de sinais associados a ameaças.

O processo de validação envolveu testes com endereços de internet maliciosos documentados no PhishTank (2025). Cada funcionalidade foi testada individualmente antes da integração no sistema completo, garantindo a qualidade dos componentes.

Por fim, realizou-se integração completa dos módulos e refinamento da interface, assegurando usabilidade para diferentes perfis de usuário e estabilidade do sistema como um todo.

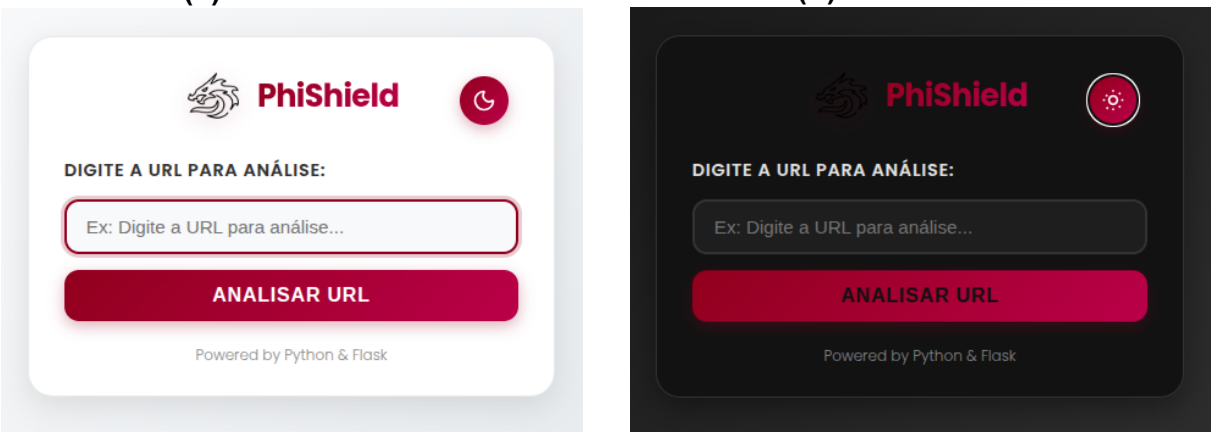
4 DESENVOLVIMENTO

4.1 Arquitetura do Sistema

O **PhiShield** adota uma arquitetura cliente-servidor organizada em três componentes principais, que trabalham em conjunto para fornecer análise de segurança em tempo real.

Figura 1 – Interface do PhiShield

(a) Modo claro **(b) Modo escuro**



Fonte: O próprio autor.

O funcionamento da extensão foi organizado em duas partes que trabalham de forma integrada: a extensão instalada no navegador e o serviço responsável pela análise das URLs. A extensão funciona como o ponto de interação com o usuário, exibindo as interfaces apresentadas na Figura 1, que podem ser visualizadas nos modos claro (a) e escuro (b). Nessa interface, o usuário insere a URL a ser verificada. Após essa etapa, a extensão envia a URL ao serviço de análise, que realiza as verificações técnicas e devolve o resultado da avaliação. Esses resultados são apresentados ao usuário de forma visual e intuitiva, permitindo a rápida compreensão do nível de risco identificado.

Além da interface visual, o sistema também disponibiliza os resultados da verificação em um formato estruturado amplamente utilizado em aplicações web, o JSON. Esse formato apresenta informações detalhadas, como a pontuação numérica atribuída, a descrição das detecções realizadas e os registros temporais da análise. Dessa maneira, ele é especialmente útil para usuários que precisam acessar toda a informação técnica de forma organizada e fácil de entender.

O serviço responsável pela análise das URLs foi desenvolvido utilizando o *framework* Flask. Ele recebe as solicitações enviadas pela extensão do navegador, realiza as verificações de segurança e devolve o resultado processado. A troca de informa-

ções entre a extensão e o serviço ocorre por meio desse mesmo formato estruturado, o que garante consistência na comunicação e permite que o retorno da análise seja entregue de maneira rápida e compreensível para o usuário

4.2 Sistema de Detecção de URLs Maliciosas

O sistema de detecção do PhiShield é composto por onze funções especializadas que analisam tanto a estrutura quanto os elementos textuais das URLs, buscando identificar padrões associados a tentativas de *phishing*. A analise inicia pela verificação do uso excessivo de hifens no domínio, já que esse caractere é frequentemente empregado em endereços maliciosos para tentar imitar sites legítimos. Em seguida, é realizada a contagem de subdomínios, permitindo identificar estruturas hierárquicas incomuns que podem indicar tentativas de camuflagem.

Durante o processo de avaliação detalhada, o sistema de detecção do PhiShield também verifica o comprimento total das URLs, considerando que endereços anormalmente longos podem representar tentativas de ofuscar o destino real. A verificação do domínio de topo (*Top-Level Domain*) observa se a extensão utilizada é frequentemente associada a atividades suspeitas. Além disso, são analisados o protocolo utilizado e os termos presentes no caminho da URL e em seus parâmetros, buscando palavras que costumam aparecer em páginas fraudulentas.

Complementando essa análise, o sistema incorpora funcionalidades para identificar encurtadores de links, detectar o uso de endereços de IP diretos, analisar a codificação de caracteres e reconhecer padrões suspeitos em subdomínios. Cada função contribui com uma pontuação específica para o *score* final, permitindo uma avaliação abrangente baseada em múltiplos indicadores de risco.

4.3 Implementações de Segurança

A solução incorpora múltiplas camadas de segurança para garantir a proteção tanto dos usuários quanto da integridade do sistema. Foi implementada uma função de sanitização de entradas que previne ataques do tipo XSS (*scripts* entre sites), por meio do escape de caracteres especiais em HTML, convertendo símbolos potencialmente maliciosos em suas respectivas “entidades seguras”.

Essa camada de proteção é aplicada em todos os pontos onde dados do usuário são renderizados na interface, assegurando que qualquer conteúdo malicioso inserido seja tratado como texto inofensivo.

Para garantir segurança e estabilidade, o sistema possui duas proteções principais: (i) primeiro, ele verifica rigorosamente o formato de cada URL, barrando endere-

ços malformados. Isso impede tanto ataques de injeção quanto de falhas inesperadas no processamento; e (ii) segundo, ele utiliza limites de tempo de resposta nas conexões, o que protege contra tentativas de sobrecarga do serviço e assegura que a extensão continue funcionando adequadamente mesmo em redes instáveis.

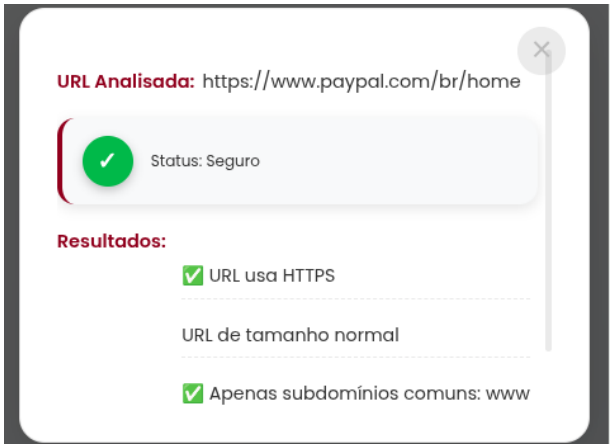
A comunicação entre a extensão e o serviço de verificação é protegida por mecanismos de validação da origem das requisições, com medidas preparadas para futura implementação de autenticação em ambiente de produção. Todas as análises são realizadas em ambiente isolado, sem persistência de dados sensíveis, garantindo a privacidade dos usuários durante o processo de verificação.

4.4 Sistema de avaliação e integração

A extensão integra um sistema de avaliação técnica com um mecanismo de comunicação estruturado, formando um fluxo contínuo desde a inserção da URL até o retorno dos resultados. O processo inicia quando o usuário digita uma URL na interface: primeiro, a entrada passa por uma filtragem inicial, incluindo validação de formato e sanitização de caracteres especiais. Em seguida, a URL é enviada ao servidor por meio de uma requisição HTTP do tipo POST, contendo os dados organizados em formato JSON (padrão amplamente utilizado para troca estruturada de informações). Essa requisição é encaminhada ao serviço de verificação identificado como `/api/check`, utilizando interface assíncrona de comunicação do JavaScript, o que mantém a interface da extensão responsiva durante o processamento.

No servidor responsável pela análise, cada uma das funções avaliadas anteriormente gera uma pontuação proporcional ao grau de risco identificado. Essas pontuações são somadas para formar um valor final que representa o nível de suspeito associado à URL. Com base nesse resultado, o sistema classifica a URL em três categorias: “seguro” quando a pontuação é igual a zero; “suspeito” quando varia entre um e três pontos; e “perigoso” quando ultrapassa quatro pontos, indicando um risco elevado de *phishing*.

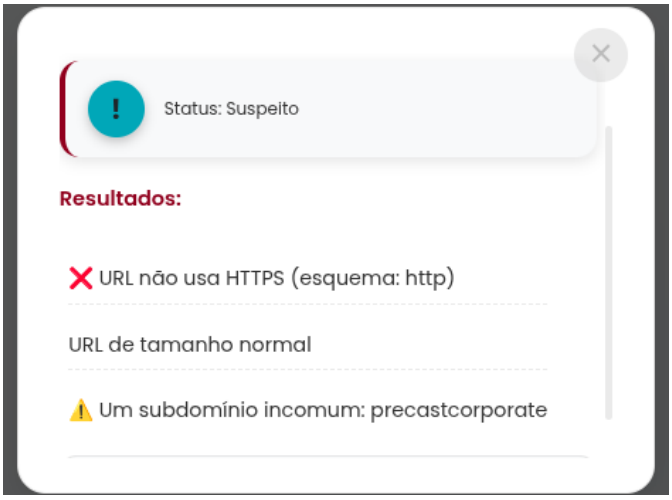
Figura 2 – Indicador de URL classificado como “seguro” (cor verde)



Fonte: O próprio autor.

Como demonstrado na Figura 2, a análise sistemática de todos os indicadores de segurança não identificou anomalias na arquitetura da URL ou em seus componentes estruturais. A conformidade com protocolos seguros de comunicação e a configuração adequada dos elementos de domínio validam a legitimidade do endereço, justificando a classificação como “seguro”, mediante o *score* zero obtido no processo de avaliação.

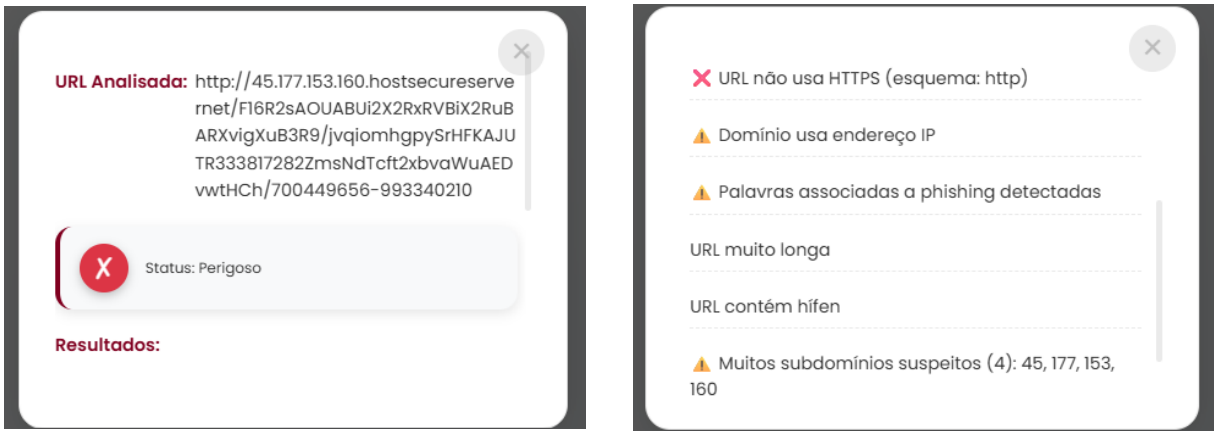
Figura 3 – Resultados detalhados quando a classificação é “Suspeito”.



Fonte: O próprio autor.

A análise da figura 3 demonstra a aplicação prática dos mecanismos de detecção implementados, evidenciando a identificação de padrões anômalos associados a páginas de *phishing*. Observa-se a presença de subdomínios incomuns e a ausência do protocolo de segurança HTTPS, fatores que contribuíram para a classificação da URL como conteúdo suspeito pelo sistema.

Figura 4 – Classificação “Perigoso” e resultados detalhados na extensão.



Fonte: O próprio autor.

As Figuras 4 demonstram a atuação do sistema de detecção diante de URLs classificadas como de alto risco. Observa-se a ausência do protocolo HTTPS combinada a outros indicadores críticos, como o uso de endereços IP no domínio e termos comumente associados a tentativas de *phishing*. A presença simultânea desses elementos reforça a eficácia do método multidimensional empregado pelo PhiShield, que avalia diferentes camadas estruturais e semânticas da URL para identificar ameaças que poderiam passar despercebidas em verificações mais superficiais.

Figura 5 – Saída em formato JSON

Fonte: O próprio autor.

Complementando a interface visual, a Figura 5 apresenta a saída dos dados em

formato JSON. Utilizada para fornecer uma visão estruturada e completa do resultado da análise. Esse retorno inclui o *score* numérico da URL, um resumo das detecções realizadas, mensagens explicativas sobre cada padrão identificado, o registro temporal da verificação (momento exato em que a análise foi executada) e a própria URL submetida. Esse formato organizado facilita tanto a interpretação técnica quanto o uso em integrações com outras aplicações.

Essa abordagem amplia a utilidade do PhiShield, pois atende a diferentes perfis de uso: usuários que necessitam apenas de uma resposta direta e desenvolvedores ou equipes técnicas que demandam dados estruturados para análises mais profundas. Dessa forma, o sistema entrega os resultados de maneira clara e completa, garantindo consistência entre a análise realizada e a forma como as informações são apresentadas.

5 CONCLUSÃO

O trabalho apresentou o desenvolvimento do PhiShield, uma extensão para navegador que implementa análise estrutural de URLs para detecção de possíveis ameaças de *phishing*. A ferramenta foi construída com módulo de interação do usuário no navegador “Google Chrome” e sistema de análise com Flask.

A arquitetura da detecção baseada em múltiplos indicadores técnicos demonstrou capacidade em flagrar ameaças sofisticadas, confirmando a efetividade do modelo analítico adotado. A interface desenvolvida oferece dois modos de apresentação: uma visualização simplificada com indicadores de cores para usuários comuns e uma saída técnica detalhada em JSON para desenvolvedores e analistas de segurança. Medidas de proteção foram implementadas para garantir a estabilidade da ferramenta durante a análise.

A abordagem desenvolvida mostrou-se promissora durante os testes realizados com bases de URLs reconhecidas. O projeto também se alinha à perspectiva de Schneier (2000), de que “a segurança é um processo, não um produto”, incorporando essa filosofia ao se apresentar como um componente evolutivo dentro do sistema

A validação realizada com bases reconhecidas permitiu testar a abordagem adotada. O trabalho identificou oportunidades, para evoluções futuras, como a integração com base de dados de ameaças em tempo real, o uso de técnicas automatizadas de aprendizados e a expansão para outros navegadores.

Em síntese o PhiShield comprova que é possível desenvolver ferramentas de segurança cibernética eficazes utilizando tecnologias acessíveis, contribuindo para a democratização da proteção digital ao oferecer uma ferramenta de código aberto.

REFERÊNCIAS

BORGES, Luiz Eduardo. **Python para Desenvolvedores'**. 2. ed. São Paulo, SP - Brasil: Novatec Editora, 2014.

CERT.BR. **Estatiticas de Phishing no Brasil**. Acesso em: 20 set. 2025. 2025. Disponível em: https://stats.cert.br/phishing/#categorias-br..

LYRA, Maurício Rocha. **Governança em Sistemas de Informação**. Editora Ciência Moderna, 2019.

MDN WEB DOCS. **HTML: Linguagem de Marcação de Hipertexto**. Acesso em: 20 out. 2025. 2024. Disponível em: https://developer.mozilla.org/en-US/docs/Web/HTML.

MDN WEB DOCS. **Javascript**. Acesso em: 20 out. 2025. 2025. Disponível em: https://developer.mozilla.org/en-US/docs/Web/JavaScript.

MDN WEB DOCS. **Trabalhando com JSON**. Acesso em: 20 nov. 2025. 2025. Disponível em: https://developer.mozilla.org/pt-BR/docs/Learn_web_development/Core/Scripting/JSON.

ORACLE. **O que é uma API?** 2025. Acesso em: 20 nov. 2025. Disponível em: https://www.oracle.com/br/cloud/cloud-native/api-management/what-is-api/.

PALLETS PROJECTS. **Flask Documentation**. Acesso em: 23 out. 2025. 2025. Disponível em: https://flask.palletsprojects.com/en/stable/.

PHISHING.ORG. **The History of Phishing**. Acesso em: 10 maio. 2025. 2021. Disponível em: https://www.phishing.org/history-of-phishing.

PHISHTANK. **PhishTank - Join the fight against phishing**. Acesso em: 5 novembro. 2025. 2025. Disponível em: https://phishtank.org/.

PROOFPOINT. **Phishing threat Reference**. 2023. Disponível em: https://www.proofpoint.com/us/threat-reference/phishing.

PROOFPOINT. **Relatório Anual de Fatores Humanos da Proofpoint revela como 2021 se tornou o ano em que os cibercriminosos se tornaram criativos**. Acesso em: 12 out. 2025. Proofpoint, Inc. 2022. Disponível em: https://www.proofpoint.com/us/newsroom/press-releases/proofpoints-annual-human-factor-report-reveals-how-2021-became-year-cyber.

PYTHON SOFTWARE FOUNDATION. **Python 3 Documentation**. Acesso em: 21 out. 2025. 2025. Disponível em: `https://docs.python.org/3/`.

SCHNEIER, Bruce. **Secrets and Lies: Digital Security in a Networked World**. 1. ed.: Wiley, 2000. ISBN 9780471253115.

TELECOM, Algar. **Segurança da informação: 10 erros ainda cometidos pelas empresas**. Acesso em: 12 out. 2025. Citação indireta de Fortinet. Blog Algar Telecom. 2024. Disponível em: `https://blog.algar.com.br/seguranca-da-informacao-10-erros-ainda-cometidos-pelas-empresas/`.

WORLD WIDE WEB CONSORTIUM. **CSS Snapshot**. Acesso em: 20 out. 2025. 2024. Disponível em: `https://www.w3.org/TR/css/`.