Universidade Federal do Estado do Rio de Janeiro
Centro de Ciências Exatas e Tecnologia
Escola de Informática Aplicada

Tupã: Scanner de vulnerabilidades

Nathan Aguiar Neves

Orientador
Fabrício Raphael Silva Pereira

Rio de Janeiro, RJ – Brasil
Março de 2023

Catalogação informatizada pelo autor

AN518

Aguiar Neves, Nathan
Tupã: Scanner de Vulnerabilidades / Nathan Aguiar Neves. -- Rio de Janeiro, 2023.
143

Orientador: Fabrício Raphael Silva Pereira.
Trabalho de Conclusão de Curso (Graduação) - Universidade Federal do Estado do Rio de Janeiro, Graduação em Sistemas de Informação, 2023.

1. automação. 2. identificação de vulnerabilidades. 3. segurança de sistemas web. 4. proteção contra ataques cibernéticos. I. Raphael Silva Pereira, Fabrício, orient. II. Título.

Tupã: Scanner de Vulnerabilidades

Nathan Aguiar Neves

Projeto de Graduação apresentado à Escola de Informática Aplicada da Universidade Federal do Estado do Rio de Janeiro (UNIRIO) para obtenção do título de Bacharel em Sistemas de Informação.

Aprovado por:

_____

Fabrício Raphael Silva Pereira (UNIRIO)

_____

Jobson Luiz Massollar da Silva

_____

Reinaldo Viana Alvares

RIO DE JANEIRO, RJ – BRASIL.

MARÇO DE 2023

Agradecimentos

Agradeço a Deus e a cada um dos mestres que me ajudaram a chegar até aqui, incluindo professores de níveis primário e secundário, amigos e também o corpo docente da Unirio. Aprendi muito com cada uma dessas pessoas, e por isso não seria justo citar nomes. Contudo, não posso deixar de citar meus maiores professores, meus pais Ademir e Márcia, com quem aprendi os valores que carrego e carregarei para o resto da vida. Além deles, agradeço ao meu orientador de projeto, o professor Fabrício Raphael Silva Pereira, por ter aceitado orientar um projeto que foge do usual do curso. Por último, dedico também este trabalho à minha namorada Juliana, que me acompanhou desde os primeiros momentos em que comecei a estudar para o Enem até a finalização deste trabalho.

RESUMO

O advento da internet permitiu a produção e o armazenamento de uma grande quantidade de informações, mas também trouxe riscos para a segurança da informação. As organizações criminosas aproveitam brechas nos sistemas para invadir entidades legítimas e obter informações valiosas. Para se protegerem, as empresas contratam profissionais de segurança ofensiva que utilizam diversas ferramentas e metodologias para identificar vulnerabilidades. Essas ferramentas possuem diferentes saídas que precisam ser analisadas pelo profissional de segurança para identificar vulnerabilidades e elaborar um relatório de segurança.

O objetivo deste trabalho foi desenvolver um sistema que automatizasse a execução e integrasse em um único sistema diversas ferramentas de descoberta de vulnerabilidades, com o fim de mapear a superfície de ataque de empresas e aumentar a eficiência dos profissionais de segurança na análise dos resultados das ferramentas.

Para cumprir o objetivo proposto, foi desenvolvido um sistema de nome Tupã, que é capaz de automatizar a execução dessas ferramentas por meio da criação de workflows e receber os resultados das análises realizadas por elas, através de suas APIs.

Por fim, a fim de verificar se o sistema desenvolvido é capaz de cumprir os objetivos propostos, o sistema foi configurado para executar algumas das ferramentas integradas nele no domínio raiz uniriotec.br, sendo capaz de encontrar 607 vulnerabilidades nos subdomínios encontrados.

Palavras-chave: automação, identificação de vulnerabilidades, segurança de sistemas web e proteção contra ataques cibernéticos.

ABSTRACT

The advent of the internet has allowed for the production and storage of a large amount of information, but it has also brought risks to information security. Criminal organizations exploit system vulnerabilities to invade legitimate entities and obtain valuable information. To protect themselves, companies hire offensive security professionals who use various tools and methodologies to identify vulnerabilities. These tools have different outputs that need to be analyzed by security professionals to identify vulnerabilities and create a security report.

The objective of this work was to develop a system that would automate the execution and integrate multiple vulnerability discovery tools into a single system in order to map the attack surface of companies and increase the efficiency of security professionals in analyzing the results of the tools.

To achieve the proposed objective, a system named Tupã was developed, which is capable of automating the execution of these tools by creating workflows and receiving the results of the analyses performed by them through their APIs.

Finally, in order to verify if the developed system is capable of fulfilling the proposed objectives, the system was configured to execute some of the integrated tools in the root domain uniriotec.br, being able to find 607 vulnerabilities in the subdomains found.

Keywords: automation, vulnerability identification, web system security, and protection against cyber attacks.

Índice

Índice de Tabela

Índice de Figuras

Lista de abreviações

- **UUID**: Identificador Único Universal (em inglês, Universally Unique Identifier).
- **TCP**: Protocolo de Controle de Transmissão (em inglês, Transmission Control Protocol).
- **UDP**: Protocolo de Datagrama de Usuário (em inglês, User Datagram Protocol).
- **SSL**: Camada de Soquetes Seguros (em inglês, Secure Socket Layer).
- **IP**: "Protocolo da Internet" (em inglês, Internet Protocol).

1.Introdução

1.1 Motivação

O advento da internet trouxe diversos benefícios para as sociedades humanas. De acordo com um relatório publicado pela fabricante de discos rígidos Seagate, nunca antes na história da humanidade produziu-se e armazenou-se tanta informação de maneira tão rápida. Junto com esse advento, governos e empresas passaram a enxergar valor nas informações. Uma outra categoria de entidade que evoluiu junto com a internet e que também percebeu o valor que essas informações possuem são as organizações criminosas, que, por sua vez, se aproveitam de brechas na implementação de sistemas para invadir entidades legítimas, de forma a se apropriarem dessas informações.

Essas invasões ocorrem por diversos motivos, sendo alguns dos mais comuns curiosidade, financeiro, notoriedade, vingança, ideologia e recreação. A dimensão do problema é tamanha que, somente em 2021, o Brasil foi o 5° país que mais sofreu ataques cibernéticos, de acordo com a consultoria alemã Roland Berger. Além disso, de acordo com a mesma empresa, somente em 2021, os prejuízos estimados podem chegar até 6 trilhões de dólares graças aos ataques cibernéticos.

Para se protegerem dessas organizações criminosas, as empresas contratam profissionais de tecnologia especializados em invadir e manter sistemas seguros. Em alguns casos, esses são encarregados de mapear todos os domínios, subdomínios, endereços de rede, portas abertas e serviços web, no intuito de simular um adversário real invadindo os sistemas da empresa que os contratou e, depois, gerar um relatório com todas essas informações. Esses elementos mapeados constituem a superfície de ataque de uma empresa e, além disso, os relatórios elencam as falhas de segurança encontradas durante o processo de acesso aos ambientes .

Existem diversas metodologias desenvolvidas por profissionais de segurança ofensiva para conseguir obter acesso a empresas. Para empregá-las, é comum utilizar uma variedade de

ferramentas que geram diferentes resultados em formatos diversos. No entanto, essa abordagem pode demandar muito tempo e esforço do profissional de segurança para analisar todas as informações coletadas e elaborar um relatório final sobre o alvo.

Diante desses problemas, percebe-se a necessidade da criação de um processo que automatize a tarefa do profissional diante de uma variedade de ferramentas e etapas. Para isso, tal processo precisa atender às seguintes necessidades:

- Automatizar a descoberta dos serviços disponibilizados online por uma determinada companhia, ajudando a diminuir serviços desconhecidos pela empresa.
- Unificar as saídas dos resultados das diversas ferramentas, facilitando a análise das informações produzidas por elas.
- Produzir relatórios de forma automática, sobrando maior tempo para os analistas de segurança realizarem outras atividades.

1.2. Objetivos

O presente trabalho tem como objetivo principal:

- Desenvolver um sistema, denominado Tupã, que seja capaz de integrar e automatizar o processo de execução de diversas ferramentas de identificação de vulnerabilidades fornecendo uma visão da superfície de ataque de uma empresa.

Para atingir o objetivo principal, esse trabalho pretende atingir os seguintes objetivos secundários:

- Elicitar requisitos de segurança de sistemas web;
- Elencar táticas, técnicas e procedimentos (TTPs) de ataques para descoberta de vulnerabilidades;
- Implementar um sistema que automatize as táticas, técnicas e TTPs elencados de forma a atingir os requisitos

1.3. Organização do texto

O presente trabalho está estruturado em capítulos e, além desta introdução, será desenvolvido da seguinte forma:

- Capítulo 2: Definições e esclarecimentos sobre alguns tópicos relacionados a esse trabalho;

- Capítulo 3: Especificações gerais do projeto, desde os requisitos da aplicação até o projeto de implementação, com diagramas;
- Capítulo 4: Detalhes do processo de construção e implementação e projeto de implementação do software;
- Capítulo 5: Ao concluir, são reunidas as considerações finais, reforçando as contribuições deste trabalho e sugere possibilidades de aprofundamento posterior.

2. Conceitos fundamentais

O objetivo deste capítulo é explicar os conceitos básicos de informação, segurança da informação, cibersegurança, *security by design hardening, workflows típicos da área de segurança, superfície de ataque*, além de enumerar e explicar algumas das principais bases de conhecimento utilizadas pela comunidade de segurança cibernética ao redor do mundo.

2.1 Informação

Segundo Buckland (1991), na língua inglesa, existem diferentes significados para a palavra informação. Isso também ocorre na língua portuguesa. A palavra informação pode ter diferentes significados a depender do contexto em que é utilizada. As mais relevantes para esse trabalho são:

1. Processo de informar. Nesse caso trata-se da ação ou o resultado desta ao informar alguém sobre algo.
2. Sinônimo de comunicação.
3. Conjunto de dados acumulados, analisados e estruturados de forma a fornecer algum valor.

De maneira geral, as informações permeiam todas as áreas da nossa vida, afetando nossas escolhas, decisões e comportamentos. O acesso a elas pode ser considerado um direito fundamental e uma condição essencial para o desenvolvimento e a evolução de indivíduos, companhias e sociedades. Segundo Humby (2006), "dados são o novo petróleo", evidenciando o valor dos dados para as companhias no paradigma econômico atual. Logo, é claro que a proteção da informação seja um fator fundamental para qualquer empresa que deseja manter sua competitividade nas áreas em que atua.

2.2 Segurança da informação

De acordo com a norma ISO-17799 (2005), a segurança da informação é definida como:

> O processo de proteção da informação contra diversos tipos de ameaças, com o objetivo de garantir a continuidade do negócio, minimizar riscos, maximizar o retorno sobre investimentos e as oportunidades de negócio.

Quando os processos de segurança não são corretamente implementados, podem surgir falhas na segurança da informação. Tais falhas são ameaças à continuidade do negócio, e podem afetar até a influência de países na geopolítica mundial.

Um exemplo de falha na segurança da informação ocorreu em 2011, quando executivos de alto escalão da Renault venderam segredos industriais para outra companhia[1] . Esse caso evidencia como a falta de segurança da informação pode levar a práticas ilícitas e prejudicar a ética e a integridade dos negócios, além de destacar a importância da adoção de medidas preventivas e de proteção de informações críticas.

2.3 Cibersegurança

Existem múltiplas definições para cibersegurança. A que será adotada para esse trabalho será a definida pela União Internacional de Telecomunicações (ITU, 2008), que define cibersegurança como:

> A coleção de ferramentas, políticas, conceitos de segurança, guias, ações, melhores práticas, treinamentos e tecnologias que podem ser utilizados para proteger o ambiente digital de uma organização e bens dela e de seus funcionários. Quando fala-se de bens inclui-se todo dispositivo digital conectado a rede, infraestrutura, aplicações, serviços e sistemas de telecomunicação, e a totalidade da informação transmitida ou armazenada no ambiente cibernético.

BUSH (2008) define que o objetivo da cibersegurança é proteger quatro princípios básicos que devem reger o ambiente cibernético e os dados contidos nele, sendo eles respectivamente: a disponibilidade, integridade, autenticidade e a confidencialidade.

A disponibilidade é o princípio que garante que as informações e os recursos estarão acessíveis sempre que necessário. Quando esse pilar é quebrado, aqueles que necessitam dessas informações e recursos podem ficar sem acesso a eles. O que pode afetar a continuidade de negócios ou a qualidade do serviço prestado.

[1] AGENCE FRANCE-PRESSE. Renault revela caso interno de espionagem grave que ameaçava ativos estratégicos. UOL Carros, São Paulo, 06 jan. 2011. Disponível em: https://www.uol.com.br/carros/noticias/afp/2011/01/06/renault-revela-caso-interno-de-espionagem-grave-e-que-ameacava-ativos-estrategicos.htm. Acesso em: 24 fev. 2023.

A integridade é o princípio que garante que as informações não foram corrompidas ou alteradas sem a permissão necessária para tal. Se esse pilar for quebrado não existe mais a confiabilidade naqueles dados, o que gera uma perda de confiança no provedor dos dados.

A autenticidade é o princípio que garante que as informações provém de uma fonte confiável. Quando esse princípio é quebrado, os usuários do serviço podem acabar caindo em golpes ou fraudes.

A confidencialidade é o princípio que garante que as informações só serão acessadas por usuários autorizados. Quando esse pilar é quebrado, informações sensíveis podem ser obtidas por usuários não autorizados.

De maneira geral, se uma empresa comete falhas ao proteger seu ambiente cibernético, acontecem incidentes como o da Equifax, que em 2017 sofreu um ataque cibernético, em que os dados de mais de 143 milhões de clientes foram expostos[2]. Este ataque deu prejuízo não apenas sobre a manutenção da estrutura cibernética da empresa, como também fez com que a empresa tivesse que pagar multas e indenizações a diversos clientes afetados.

2.4 Security by design

De acordo com Santos (2007), *Security by design* consiste na noção de que um sistema de software deve ser pensado desde o início para ser, em tese, à prova de violações de segurança buscando garantir a disponibilidade, integridade, autenticidade, além da confidencialidade do sistema e dos dados contidos nele.

[2] MASSIVE Equifax data breach hits 143 million. BBC, [*S. l.*], p. 1, 8 set. 2017. Disponível em: https://www.bbc.com/news/business-41192163. Acesso em: 24 fev. 2023.

2.5 Hardening

Segundo Barker (2014) "*Hardening*" (traduzido de maneira literal para o português como "endurecimento") é o conjunto de técnicas e práticas que visam diminuir ao máximo a superfície de ataque de uma aplicação. Diferente do *Security by Design* que é aplicado muito antes no ciclo de vida da aplicação, as técnicas de *Hardening* costumam ser aplicadas quando a aplicação está para ir ou já está em ambiente de produção.

2.6 Workflow típico de segurança de Informação

Para compreender este trabalho, é necessário definir dois conceitos. Embora ambos existam, Workflow é um termo mais comumente utilizado no mercado, enquanto Scan é utilizado tanto no mercado quanto na academia:

- **Workflow**: uma sequência de execução de tarefas escrita como código para ser executada de forma automatizada.
- **Scan**: O significado original de Scan ou Scanning de acordo com CNSS-4009 (2022) significa "Enviar pacotes ou solicitações para outro sistema para obter informações que serão usadas em um ataque subsequente.”. Contudo, a definição que utilizaremos neste trabalho será o processo de execução de um workflow em um alvo previamente configurado. Com o fim de realizar uma análise da superfície de ataque do alvo, em busca de subdomínios, portas, endereços de ip, tecnologias e vulnerabilidades.

2.7 Superfície de ataque

Podemos usar uma definição semelhante a Chaplain (2018) como o conjunto de qualquer componente de um sistema, elementos ou ambientes em que um atacante pode utilizar para invadir, causar efeito inesperado ou extrair informação de um sistema. Ela deve ser remapeada de tempos em tempos, pois dessa maneira é possível identificar recursos vulneráveis que poderiam ser utilizados por criminosos e realizar a correção dessas vulnerabilidades.

2.8 Bases de conhecimento e metodologias utilizadas pela comunidade de cibersegurança

Existem diversas bases de dados e de conhecimento utilizadas pela comunidade de cibersegurança para diversos fins, em que se mapeia as experiências já vividas pela comunidade, visando mitigar as falhas de segurança já conhecidas, ou evitar que novas surjam. Algumas das bases mais importantes são:

- **CVE[3]:** Sistema de indexação de vulnerabilidades que fornece um identificador para cada vulnerabilidade descoberta, criando uma maneira de duas ferramentas, processos ou pessoas saberem que estão falando da mesma vulnerabilidade.
- **CWE[4]**: Base de dados comunitária que tem como objetivo listar os diferentes tipos de vulnerabilidades.
- **MITRE PRE-ATT&CK®[5]:** Base de conhecimento com técnicas, táticas e procedimentos baseadas na observação de Ameaças Persistentes Avançadas. Cada técnica, tática e procedimento possui um nome e um ID para localização. Sendo uma tática um conjunto de técnicas, para esse trabalho utilizaremos apenas as técnicas pertencentes à tática de reconhecimento(*Reconnaissance*). Sendo essas as diversas técnicas utilizadas por atacantes para ganhar informação que posteriormente pode ser utilizada para conduzir um ataque ao alvo.
- **Cyber Killchain[6]:** é uma metodologia desenvolvida pela Lockheed Martin utilizada para descrever, de maneira superficial, as diferentes etapas de um ataque cibernético, desde o reconhecimento até a exploração e exfiltração de dados. Ela é baseada em uma abordagem sequencial e linear, onde cada etapa depende da anterior para que o ataque seja bem-sucedido.

[3] CVE. FAQs. Disponível em: https://www.cve.org/ResourcesSupport/FAQs#pc_introcve_nvd_relationship. Acesso em: 27 fev. 2023.
[4] MITRE Corporation. CWE - Common Weakness Enumeration. Disponível em: https://cwe.mitre.org. Acesso em: 27 fev. 2023.
[5] MITRE Corporation. ATT&CK Matrix for Enterprise - Pre-Attack. Disponível em: https://attack.mitre.org/matrices/enterprise/pre/. Acesso em: 27 fev. 2023
[6] LOCKHEED MARTIN. Cyber Kill Chain®. Disponível em: https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html. Acesso em: 05 mar. 2023.

3. Trabalhos Relacionados

O objetivo deste capítulo é apresentar outros trabalhos relacionados a esse trabalho, realizar uma comparação entre eles, quando pertinente, e sua relação com a proposição deste trabalho.

Para fins de comparação entre este e outros trabalhos, utilizaremos os seguintes critérios que correspondem a fases típicas de um workflow de segurança da informação:

- **Enumera Subdomínios**: capacidade de identificar subdomínios de um determinado domínio raiz, ou seja, listar os subdomínios que estão associados ao domínio principal.
- **Identifica Portas**: capacidade de detectar quais portas estão abertas em um determinado host, o que pode ajudar na identificação de possíveis vulnerabilidades ou serviços em execução.
- **Identifica Vulnerabilidades**: capacidade de identificar falhas em sistemas ou aplicações.
- **Identifica Endpoints**: Refere-se à capacidade da ferramenta de identificar quais são os pontos de entrada (endpoints) em uma aplicação, o que, por sua vez, ajuda a mapear a aplicação.
- **Identifica Tecnologias**: capacidade de descobrir quais tecnologias e *frameworks* de desenvolvimento estão sendo utilizados em uma aplicação, o que pode ajudar na procura por vulnerabilidades focadas para aquela tecnologia.
- **Elabora relatório**: capacidade da ferramenta de gerar um relatório completo com as informações coletadas durante os processos listados anteriormente.

Foi decidido utilizar o MITRE PRE-ATT&CK em vez da Cyber Killchain como metodologia, devido às informações mais detalhadas que ele fornece sobre o funcionamento do processo de um ataque a um alvo, em comparação com a visão geral oferecida pela Cyber Killchain. A tabela 1 apresenta a relação entre cada critério e a respectiva técnica, identificador e descrição da técnica presente no MITRE PRE-ATT&CK (2020).

Tabela 1: Tabela de relação entre os critérios de avaliação das ferramentas e as técnicas presentes na tática de reconhecimento(Reconnaissance[7]) presentes no MITRE PRE-ATT&CK.

Critério	Técnica	Código da técnica	Descrição da técnica
Enumera Subdomínios	Coletar informações de rede da vítima: DNS.	T1590/002	Adversários podem fazer varreduras em vítimas em busca de vulnerabilidades que podem ser usadas durante o direcionamento. As varreduras de vulnerabilidades geralmente verificam se a configuração de um host/aplicativo de destino (por exemplo, software e versão) potencialmente se alinha com o alvo de uma exploração específica que o adversário pode procurar usar.
Identifica Portas	Coletar informações de rede da vítima: Endereços IP.	T1590/005	Adversários podem coletar os endereços IP da vítima que podem ser usados durante o direcionamento do ataque. Os endereços IP públicos podem ser alocados para organizações por bloco ou uma sequência de endereços. As informações sobre os endereços IP atribuídos podem incluir uma variedade de detalhes, como quais endereços IP estão em uso. Os endereços IP também podem permitir que um adversário derive outros detalhes sobre a vítima, como tamanho da organização, localização física(ões), provedor de serviços de Internet e onde/como sua infraestrutura pública está hospedada.

[7] MITRE CORPORATION. MITRE ATT&CK. Disponível em: https://attack.mitre.org/tactics/TA0043/. Acesso em: 28 fev. 2023.

Identifica Endpoints	Escaneamento Ativo: Varredura de *wordlists*	T1595/003	Os adversários podem sondar iterativamente a infraestrutura usando técnicas de brute-forcing e crawling. Embora essa técnica empregue métodos semelhantes aos de Brute Force, seu objetivo é identificar conteúdo e infraestrutura em vez de descobrir credenciais válidas. As listas de palavras usadas nessas varreduras podem conter nomes genéricos e extensões de arquivo comumente usadas ou termos específicos de um software específico. Os adversários também podem criar listas de palavras personalizadas para o alvo usando dados coletados de outras técnicas de Reconhecimento (por exemplo, Coletar informações da organização da vítima ou Pesquisar sites de propriedade da vítima).
Identifica Vulnerabilidades	Escaneamento Ativo: Varredura de Vulnerabilidades.	T1595/002	Os adversários podem escanear as vítimas em busca de vulnerabilidades que podem ser usadas durante o direcionamento do ataque. Os escaneamentos de vulnerabilidades geralmente verificam se a configuração de um host/aplicação de destino (por exemplo: software e versão) potencialmente se alinha com o alvo de um exploit específico que o adversário pode querer usar.

3.1 Nuclei

O Nuclei[8] É uma ferramenta de código aberto criada pela empresa project discovery para varredura de vulnerabilidades em aplicações web. Foi escrita em Go e é conhecida pela sua facilidade e velocidade em sua execução.

Para realização de análises o Nuclei depende de arquivos no formato YAML que contêm regras sobre quais requisições serão feitas e quais respostas são esperadas no caso da confirmação de uma vulnerabilidade.

Figura 1: Exemplo de arquivo YAML utilizado pelo Nuclei

```yaml
id: git-config

info:
  name: Git Configuration - Detect
  author: pdteam,pikpikcu,Mah3Sec_
  severity: medium
  description: Git configuration was detected via the pattern /.git/config and log file on passed URLs.
  classification:
    cvss-metrics: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N
    cvss-score: 5.3
    cwe-id: CWE-200
  tags: config,git,exposure

requests:
  - method: GET
    path:
      - "{{BaseURL}}/.git/config"

    matchers-condition: and
    matchers:
      - type: word
        words:
          - "[core]"

      - type: dsl
        dsl:
          - "!contains(tolower(body), '<html')"
          - "!contains(tolower(body), '<body')"
        condition: and

      - type: status
        status:
          - 200
```

Fonte: PROJECTDISCOVERY. Nuclei Templates: Exposures Configs - git-config.yaml. Disponível em: https://github.com/projectdiscovery/nuclei-templates/blob/main/exposures/configs/git-config.yaml. Acesso em: 21 mar. 2023.

Seu maior ponto forte é a grande quantidade de vulnerabilidades que cobre, uma vez que os templates utilizados por ele são mantidos pela comunidade. É possível, em poucos minutos, realizar testes para identificação de centenas de vulnerabilidades.

[8] PROJECT DISCOVERY. Nuclei. GitHub, [2019]. Disponível em: https://github.com/projectdiscovery/nuclei. Acesso em: 24 fev. 2023.

Porém, sua principal limitação é que ele não realiza análises mais profundas do que foi encontrado. Além disso, por ser baseado em templates, a identificação de vulnerabilidades de ordem mais complexa acaba ficando limitada.

3.2 Nmap

Outra ferramenta bastante conhecida no mundo da segurança da informação é o Nmap. Desenvolvida em C e lançada em 1997, o Nmap é uma ferramenta de código aberto que permite a varredura de redes e identificação de hosts, portas tcp e udp, serviços e sistemas operacionais.[9]

Para realizar o scan, o Nmap utiliza técnicas de varredura de portas, verificação de protocolos e identificação de serviços para mapear a rede e identificar possíveis vulnerabilidades. A ferramenta permite que o usuário personalize as varreduras de acordo com suas necessidades utilizando uma série de parâmetros no momento da execução da formatura.

Entre seus pontos fortes, podemos citar sua flexibilidade e configuração, a sua capacidade de realizar varreduras rápidas e a sua ampla gama de opções e funcionalidades. Além disso, o Nmap possui uma grande comunidade de usuários e desenvolvedores, o que garante uma constante atualização da ferramenta.

No entanto, o Nmap também apresenta alguns pontos fracos, como a possibilidade de gerar falsos positivos e falsos negativos, especialmente na detecção de portas tcp e na detecção do sistema operacional. Além disso, a ferramenta pode ser detectada por alguns sistemas de detecção de intrusão e antivírus, o que pode limitar seu uso, caso ela seja executada dentro do ambiente interno de um alvo.

3.3 Subfinder

Assim como o Nuclei, o Subfinder[10] Foi desenvolvido pela empresa Project Discovery e é uma ferramenta de código aberto escrita em Go que permite a identificação de subdomínios de um determinado alvo.

Para encontrar subdomínios, o Subfinder utiliza técnicas de força-bruta e verificação de DNS para identificar subdomínios que possam estar vinculados ao alvo. A ferramenta

[9] NMAP. Nmap Documentation. [S.l.], [2021]. Disponível em: https://nmap.org/docs.html. Acesso em: 24 fev. 2023.
[10]PROJECT DISCOVERY. Subfinder. Repositório GitHub, [2021]. Disponível em: https://github.com/projectdiscovery/subfinder. Acesso em: 24 fev. 2023.

também permite que o usuário personalize as varreduras de acordo com suas necessidades, sendo possível realizá-la de maneira passiva (Apenas verificação de DNS), Ativa (Tentando encontrar subdomínios através de força-bruta) e das duas maneiras, também é possível configurar de quais fontes a verificação passiva utilizará como base.

Entre seus pontos fortes podemos citar a sua capacidade de identificar subdomínios que podem estar escondidos ou não divulgados publicamente, o que pode aumentar a superfície de ataque do alvo. Além disso, a ferramenta é rápida e eficiente, permitindo que o usuário obtenha resultados em poucos minutos.

Todavia, o Subfinder também apresenta limitações, como a possibilidade de gerar falsos positivos e falsos negativos na detecção de subdomínios, especialmente no modo Ativo. Além disso, seu maior defeito é que a ferramenta não é capaz de identificar vulnerabilidades em si, limitando-se apenas a identificar subdomínios.

3.4 Burp Suite

Diferente do Subfinder e do Nuclei, desenvolvido pela Portswigger, o Burp Suite é uma plataforma completa para testes de penetração em aplicações web.[11]

O Burp Suite possui uma ampla gama de recursos que permitem a identificação de vulnerabilidades na web, servindo como um "canivete suiço", com ferramentas para interceptação de tráfego, fuzzing e em suas versões professional e enterprise, um scanner de vulnerabilidades web.

Entre seus pontos fortes, podemos citar a sua capacidade de analisar aplicações web complexas e suas integrações através de plugins construídos pela comunidade de usuários com outras ferramentas para auxiliar nos testes de segurança, tais como o Nmap e o Metasploit.

Contudo, o Burp Suite possui limitações, como a necessidade de treinamento adequado para sua utilização efetiva. Além disso, o software pode ser considerado pesado e consome muitos recursos de memória e processamento.

3.5 Acunetix

O Acunetix[12] é uma ferramenta de segurança da informação projetada para verificar vulnerabilidades em aplicações web e em sites. Possui uma interface gráfica amigável e

[11] Link para o site da PortSwigger: https://portswigger.net/burp
[12] Site do Acunetix: https://www.acunetix.com/

funcionalidades robustas para varredura de vulnerabilidades, gerenciamento de relatórios e configuração de políticas de segurança.Além disso, o Acunetix é conhecido por sua capacidade de varredura rápida e precisa de aplicações web, reduzindo o tempo necessário para identificar e corrigir vulnerabilidades.

No entanto, a principal limitação do Acunetix é seu alto custo, que começa a partir de 4000 dólares por website[13], o que pode torná-lo impraticável para pequenas empresas e projetos individuais.

3.6 OpenVAS

O OpenVAS[14] é uma ferramenta de código aberto para varredura em redes e sistemas. Ele é projetado para verificar se os hosts em uma rede estão seguros e configurados corretamente.

Entre as funcionalidades do OpenVAS, podemos destacar sua capacidade de detectar vulnerabilidades comuns nos recursos da rede interna, como falhas de configuração, vulnerabilidades ocasionadas por software desatualizado e problemas de segurança de rede. O OpenVAS também oferece suporte a protocolos comuns de rede, como SNMP e SSH.

Além disso, o OpenVAS permite que os usuários personalizem suas varreduras de acordo com suas necessidades. O OpenVAS também é escalável, permitindo a varredura de grandes redes e sistemas complexos.

Entretanto, o OpenVAS possui alguns problemas em seu uso, sendo esses a sua complexidade no momento da instalação e da configuração da varredura. Além disso, a qualidade dos resultados de varredura pode ser afetada pela configuração inadequada ou pelo uso incorreto da ferramenta.

A tabela 2 trata-se de uma matriz de comparação de algumas das principais ferramentas disponíveis hoje com o objeto deste trabalho. Os critérios de comparação são os elicitados anteriormente.

Para facilitar na formatação da tabela utilizaremos uma legenda para cada um dos critérios, sendo esses:

- Enumera Subdominios (**ES**)
- Identifica Portas (**IP**)
- Identifica Vulnerabilidades (**IV**)

[13]Acesso em: 24 fev. 2023. Disponível em: https://www.saasworthy.com/product/acunetix/pricing.
[14] OPENVAS. OpenVAS. Disponível em: https://www.openvas.org/. Acesso em: 24 fev. 2023.

- Identifica Endpoints (**IE**)
- Identifica Tecnologias (**IT**)
- Elabora Relatório (**ER**)

Tabela 2: Tabela de comparação entre as diversas ferramentas utilizadas no mercado e o resultado deste trabalho

Ferramenta	ES	IP	IV	IE	IT	ER
Nuclei	Não	Não	Sim	Parcial	Sim	Não
Nmap	Não	Sim	Sim	Não	Sim	Não
Subfinder	Sim	Não	Não	Não	Não	Não
Burp Suite	Não	Não	Pago	Pago	Sim	Pago
Open VAS	Não	Sim	Sim	Não	Sim	Sim
Acunetix	Não	Não	Sim	Sim	Sim	Sim
Tupã	**Sim**	**Sim**	**Sim**	**Sim**	**Sim**	**Sim**

3.7 Automatização de um processo de varredura

Diante das informações apresentadas na seção anterior, é possível notar que embora todas as ferramentas elencadas atendam a alguns critérios da tabela, nenhuma delas é capaz de atender a todos eles. Portanto, é necessário que o profissional de segurança utilize uma combinação de ferramentas para conseguir completar todos os critérios listados na seção anterior, o que por sua vez, causa o problema descrito no início deste trabalho onde, para gerar um relatório, um profissional de segurança precisaria analisar o resultado de várias ferramentas. Eliminar as repetições e só então, manualmente, elaborar um relatório com base na saída dessas ferramentas.

O resultado deste trabalho buscará sanar esses problemas através da orquestração da execução de ferramentas de segurança e de uma lógica que receberá e tratará a saída dessas ferramentas.

4. Especificações do Sistema Tupã

O objetivo deste capítulo é definir as especificações do projeto antes da fase de implementação do software, passando pelos requisitos funcionais, não funcionais, público alvo, diagrama de casos de uso e estrutura do banco de dados.

4.1 Proposta: O scanner Tupã

O Tupã foi criado com o objetivo de fornecer para os times de segurança, uma visão geral sobre a superfície de ataque de uma empresa, mostrando em uma aplicação web, os subdomínios, portas abertas nos hosts, vulnerabilidades e tecnologias utilizadas. Além de servir para centralizar o conhecimento do time de segurança ofensiva a respeito dos alvos.

4.1.1 Público Alvo

O Tupã é destinado a profissionais de segurança da informação, desenvolvedores preocupados com a segurança de sua aplicação, empresas, órgãos governamentais e não governamentais.

4.1.2 Benefícios

Dentre os benefícios técnicos de sua utilização, é possível citar:

1. A visualização da superfície de ataque do alvo, ajudando a eliminar o *Shadow It*[15].
2. A identificação de vulnerabilidades em tempo real através da configuração de alarmes.
3. A possibilidade de geração de relatórios com base nos resultados das ferramentas utilizadas no scan.

[15] Shadow It: Qualquer recurso de tecnologia que existe, porém não está inventariado pela companhia. Ex: Um site de teste que um desenvolvedor colocou online, mas que por algum motivo, não avisou aos times de infraestrutura e segurança sobre a sua existência

4.2 Requisitos Funcionais

Para que a aplicação seja capaz de resolver o problema proposto, foram definidos os seguintes requisitos funcionais:

1. Criar perfis de usuários de maneira modular.
2. Fornecer uma forma dos usuários gerenciarem os workflows.
3. Permitir que os usuários tenham uma boa visualização da superfície de ataque do alvo.
4. Permitir que os usuários gerem relatórios sobre tudo que for possivelmente encontrado em um scan.
5. Permitir que scans possam ser agendados e executados automaticamente.
6. Permitir que os usuários possam comparar o que foi encontrado entre dois scans em um mesmo alvo.

4.3 Requisitos não funcionais

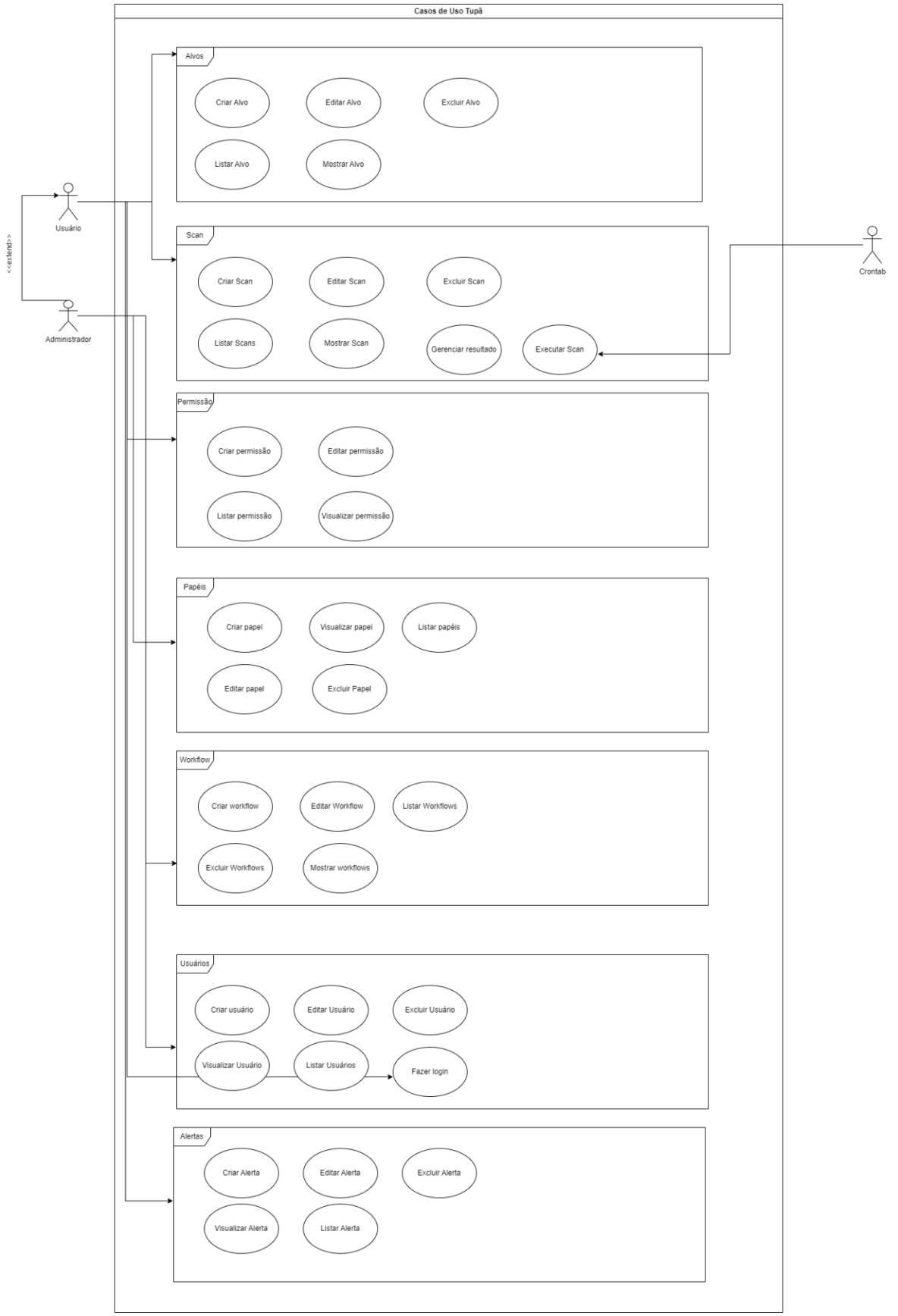
Para garantir a qualidade do produto final a ser desenvolvido e a boa utilização dos seus usuários, a aplicação deve seguir os seguintes requisitos não funcionais:

1. Deve ficar disponível 24/7.
2. A resposta das páginas não pode ser maior que 20 segundos.
3. Toda a infraestrutura deve estar virtualizada (contêineres) de modo a proteger a máquina que vai hospedá-la e reiniciar a aplicação rapidamente em caso de interrupção do serviço.

4.2 Casos de Uso

A partir dos requisitos funcionais, 38 casos de uso principais foram definidos, sendo que com exceção de realizar o scan e gerenciar os resultados do scan, cada um deles compreende as 4 operações relacionadas ao CRUD (*CREATE*, *READ*, *UPDATE* e *DELETE*) esses podem ser vistos no diagrama de casos de uso presente na figura 2, e os 30 casos de uso mais importantes para o funcionamento do Tupã descritos com mais detalhes ao longo das próximas seções deste capítulo.

Figura 2: Resumo do diagrama de casos de uso



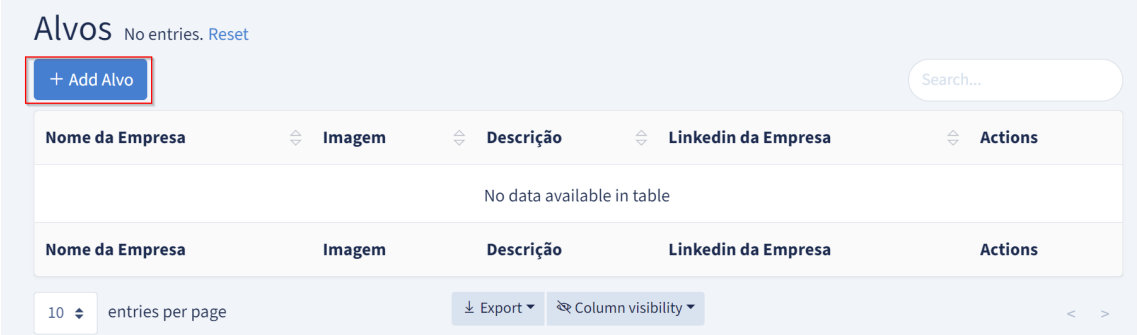
4.2.1 Criar Alvo

Esse caso de uso permite a criação de um alvo para a realização do scan

4.2.1.1 Fluxo Básico

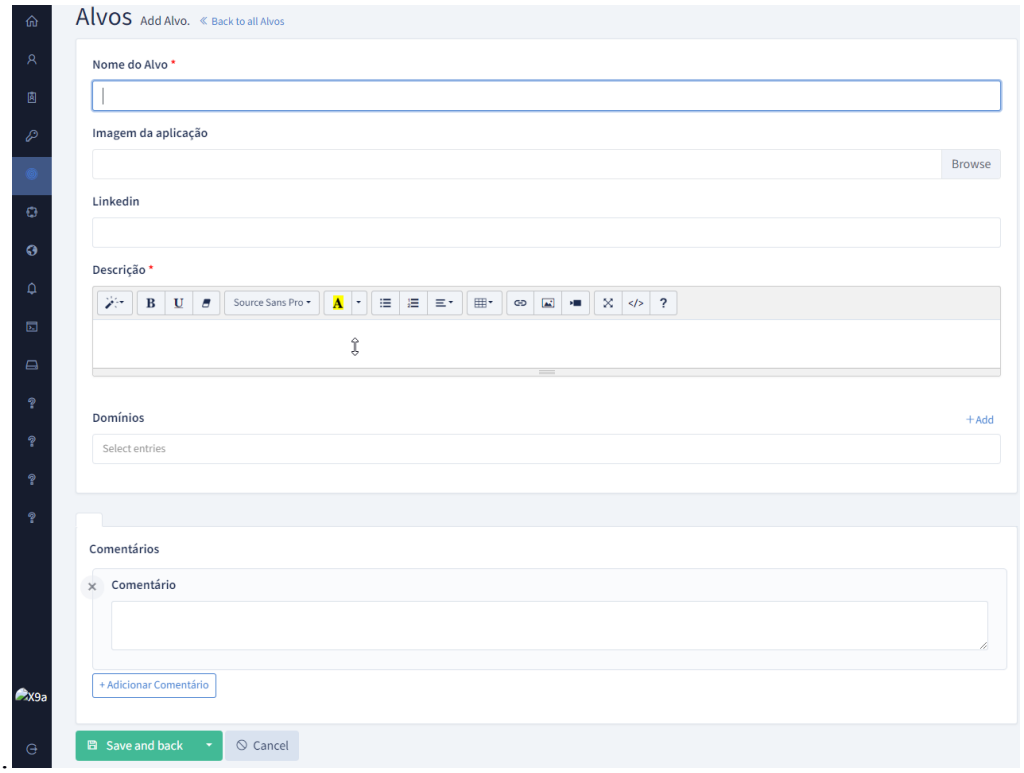
1. O usuário seleciona a opção criar alvo em “listar alvos”.

Figura 3: “ Listar Alvos “



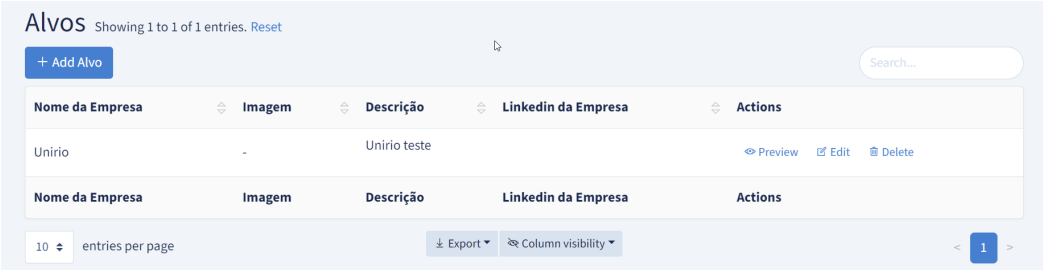
2. O usuário informa nome, descrição, LinkedIn (se houver) e domínios relacionados

Figura 4: Informar os dados do usuário



3. Usuário confirma, recebe a mensagem que o alvo foi criado com sucesso e volta a página de listar alvos.

Figura 5: Usuário confirma e recebe mensagem que o alvo foi criado



4.2.1.2 Pré Condições

1. O usuário precisa estar autenticado.
2. O usuário precisa estar na tela de listar alvos

4.2.1.3 Pós condições

1. Novo alvo é adicionado ao sistema

4.2.1.4 Restrições

1. O alvo deve ter pelo menos um domínio relacionado
2. O alvo deve ter uma descrição
3. O nome do alvo precisa ter pelo menos 5 caracteres
4. O usuário precisa possuir a permissão de modificar alvos ou ser administrador do sistema

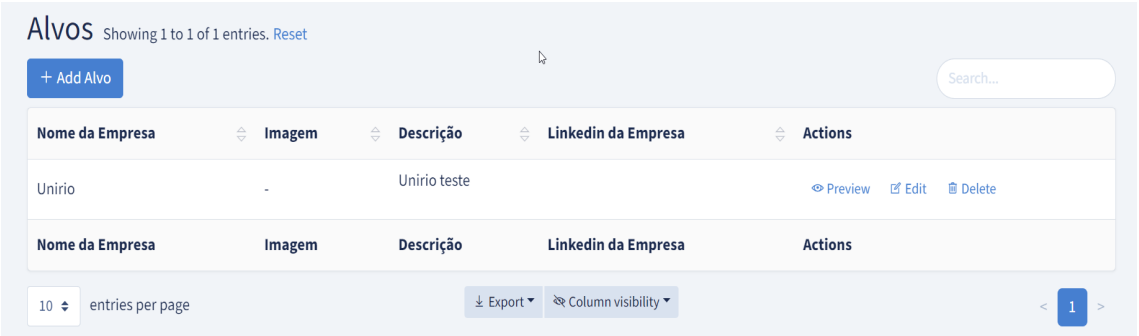
4.2.2 Excluir Alvo

Este caso de uso permite a possibilidade de excluir um alvo previamente cadastrado no Tupã

4.2.2.1 Fluxo Básico

1. O usuário habilitado acessa a lista de alvos

Figura 6: Acessar a lista de alvo

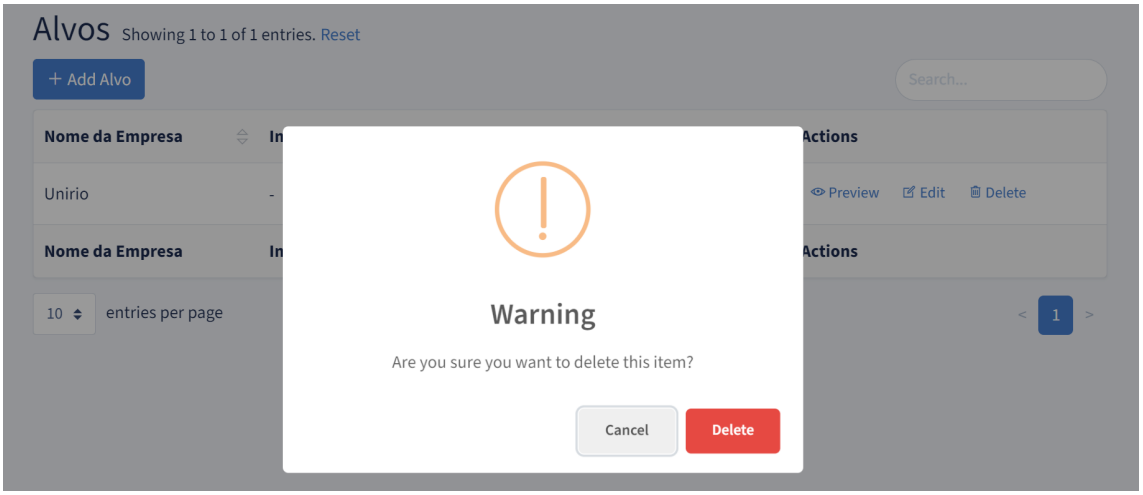


2. O usuário habilitado seleciona o botão delete

Figura 7: Selecionar o botão delete

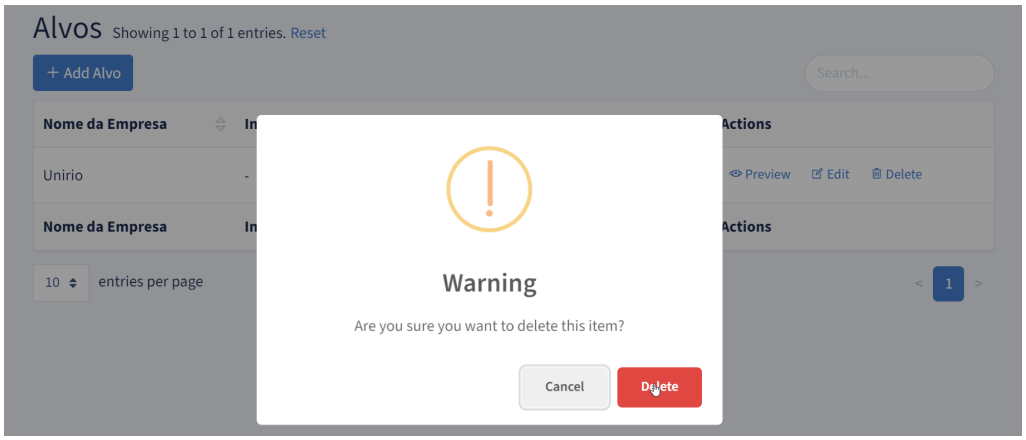
3. O sistema exibe uma mensagem de confirmação para o usuário.

Figura 8: Exibição da mensagem de confirmação para o usuário



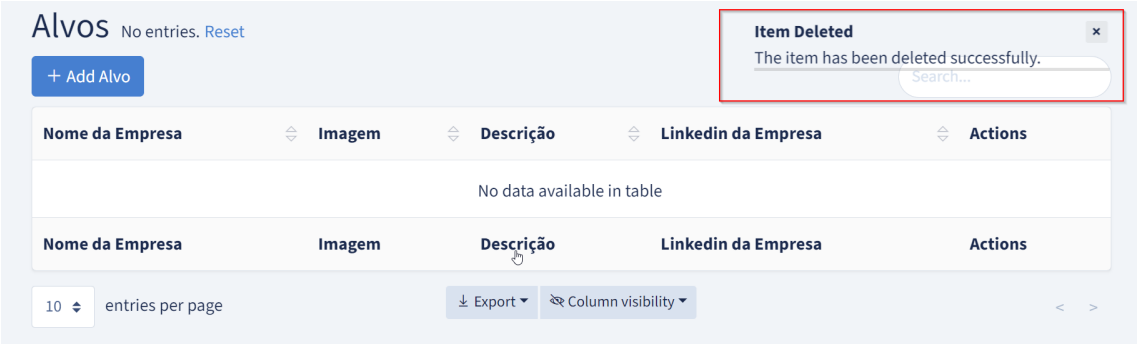
4. O usuário confirma a exclusão do alvo

Figura 9: Confirmação da exclusão do alvo



5. O sistema exibe uma mensagem de sucesso informando que o alvo foi excluído com sucesso.

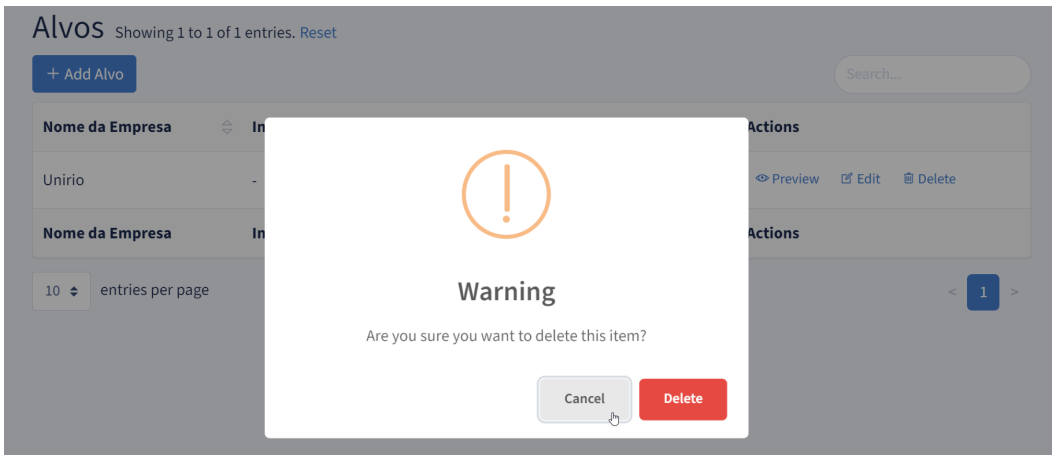
Figura 10: Exibição com a informação que o alvo foi excluído



4.2.2.2 Fluxo Alternativo

1. Se o usuário desistir da exclusão do alvo, ele pode cancelar a operação e voltar para a página de listar alvos.

Figura 11: Caso o usuário desista da exclusão, pode-se clicar na opção " cancel"



4.2.2.3 Pré-condições

1. Existe pelo menos um alvo cadastrado no sistema.
2. O usuário possui permissão para excluir alvos.

4.2.2.4 Pós-condições

1. O alvo selecionado é excluído do sistema.
2. O sistema volta para a página de listar alvos.

4.2.2.5 Restrições

1. O usuário só pode excluir alvos que ele mesmo criou.
2. Para excluir um alvo é necessário deletar todos os scans atrelados a ele

4.2.3 Editar Alvo

Este caso de uso permite a edição das informações de um alvo previamente cadastrado no sistema Tupã.

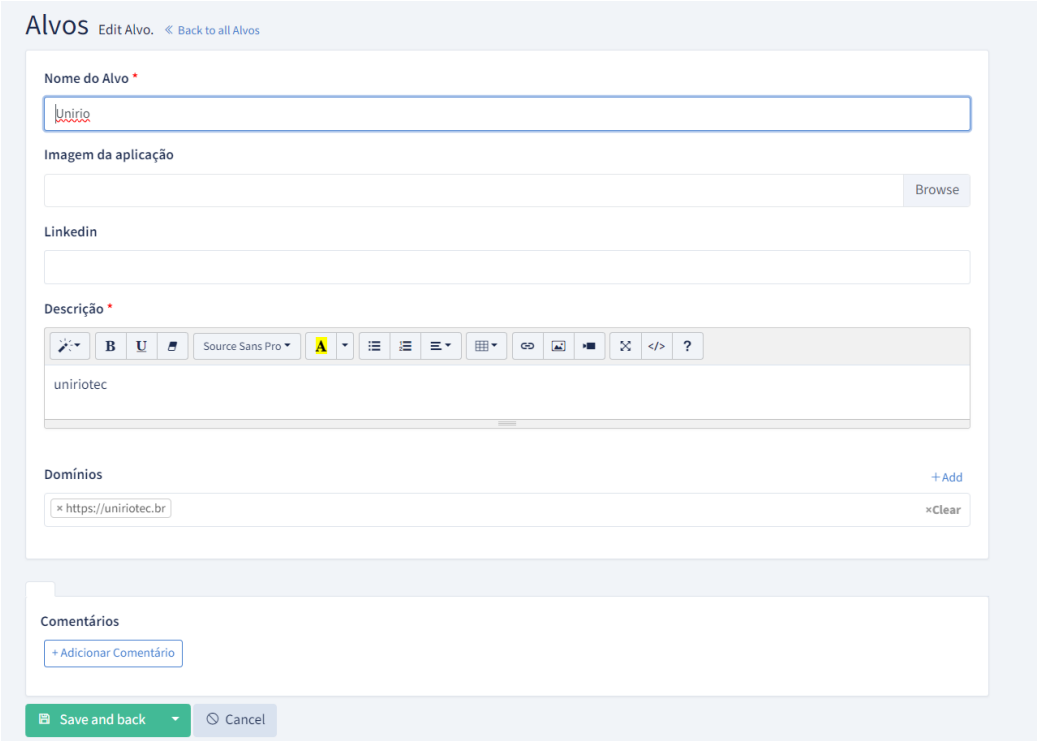
4.2.3.1 Fluxo Básico

1. O usuário habilitado seleciona a opção de editar alvo na página onde os alvos são listados.

Figura 12: Página onde os dados são listados

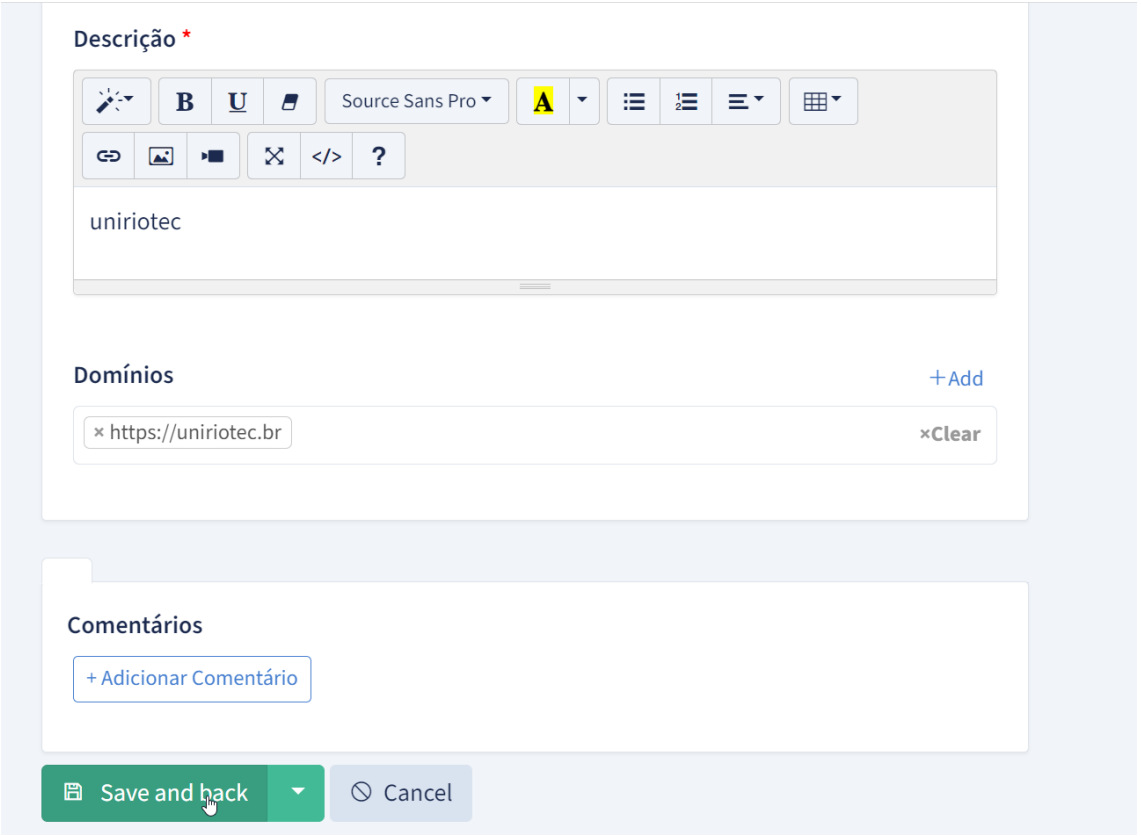
2. O usuário habilitado modifica as informações do alvo, como nome, descrição, LinkedIn e domínios relacionados.

Figura 13: Página onde o usuário pode alterar informações



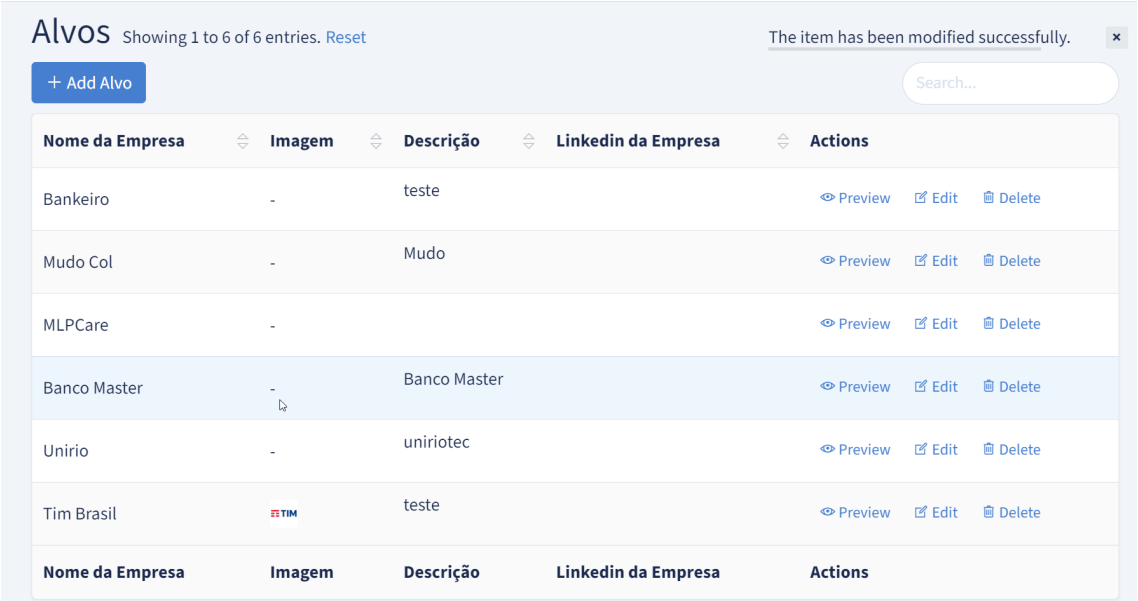
3. O usuário confirma a alteração realizada.

Figura 14: Confirmação da alteração realizada



4. O sistema atualiza as informações do alvo.
5. O usuário é redirecionado para a página de “ listar alvos “.

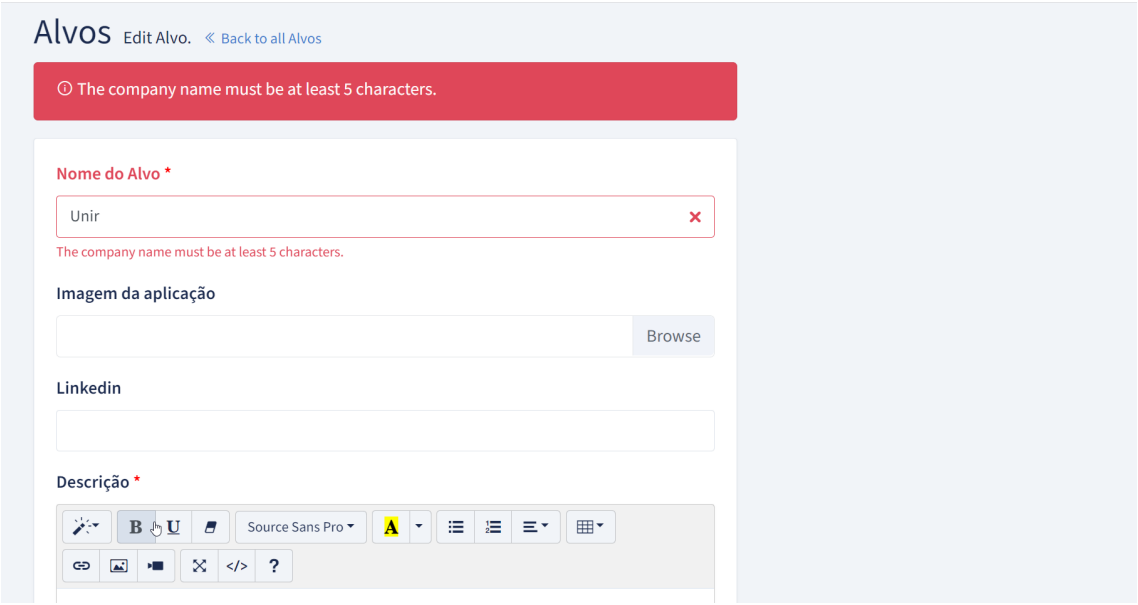
Figura *15: Redirecionamento para a página de “ listar alvos”*



4.2.3.2 Fluxo Alternativo

1. O usuário modifica as informações do alvo com dados inválidos.

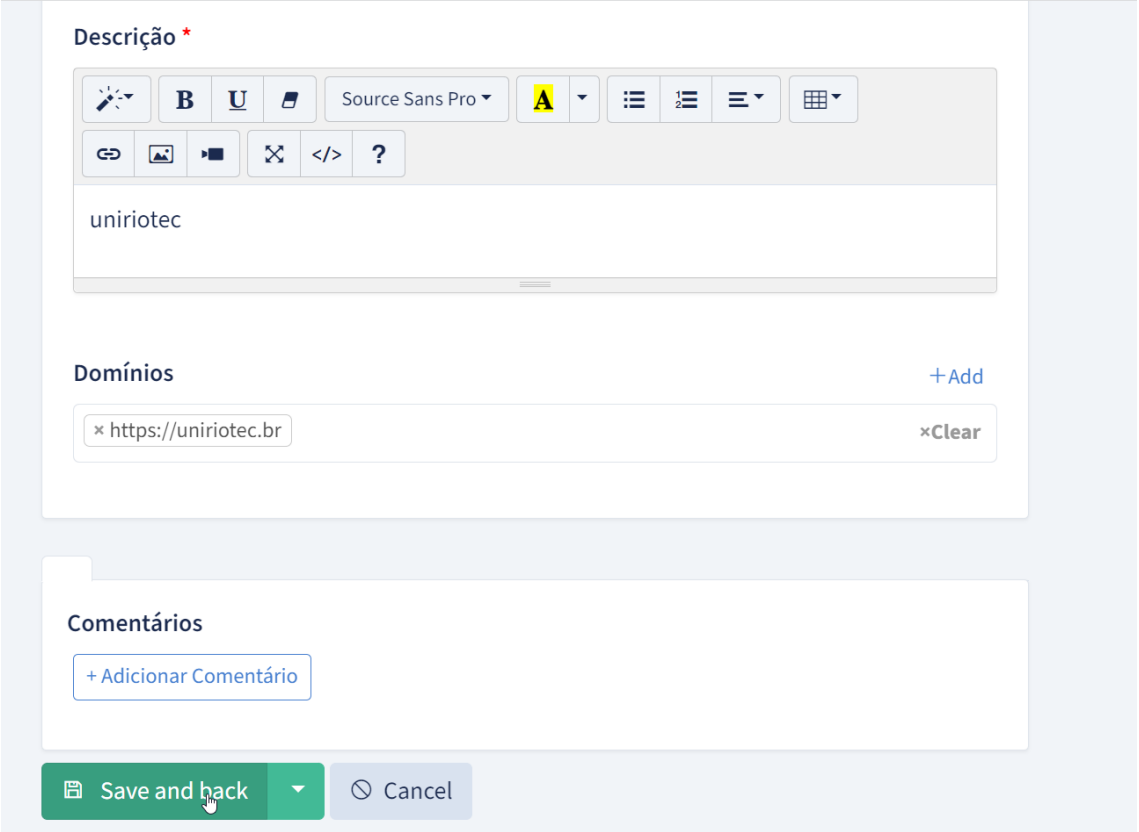
Figura 16: Apresentação de dados inválidos



2. O sistema emite uma mensagem de erro informando que os dados são inválidos e solicita ao usuário que insira informações válidas.

3. O usuário realiza a correção dos dados e confirma a alteração novamente.

Figura 17: Correção de dados e confirmação de alterações



4.2.3.3 Pré-condições

1. O usuário deve estar autenticado e habilitado a editar alvos.

4.2.3.4 Pós-condições

1. As informações do alvo selecionado são atualizadas no sistema.

4.2.3.5 Restrições

1. O alvo deve ter pelo menos um domínio relacionado.
2. O alvo deve ter uma descrição.
3. O nome do alvo precisa ter pelo menos 5 caracteres.

4.2.4 Listar Alvos

Este caso de uso permite que o usuário visualize a lista de todos os alvos cadastrados no sistema Tupã.

4.2.4.1 Fluxo Básico

1. O usuário habilitado seleciona a opção "Listar Alvos".

Figura 18: Selecionar a opção "Listar Alvos"

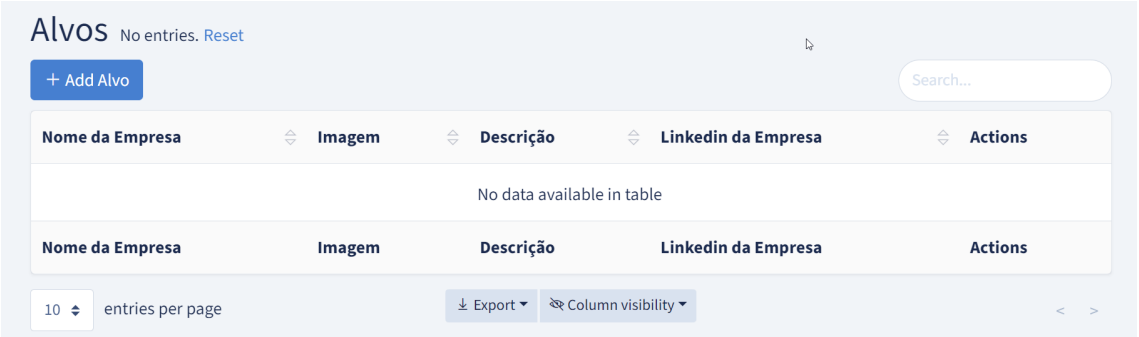
2. O sistema exibe uma tabela contendo todos os alvos cadastrados, mostrando informações como nome, descrição e domínios relacionados.

Figura 19: Exibição de informações

4.2.4.2 Fluxo Alternativo

1. Se não houver nenhum alvo cadastrado, o sistema exibe uma mensagem informando que não existem alvos cadastrados no momento.

Figura 20: Sistema informa que não existe alvos cadastrados.



4.2.4.3 Pré-condições

1. O usuário precisa estar logado no sistema Tupã.
2. Existem alvos cadastrados no sistema.

4.2.4.4 Pós-condições

1. O usuário visualiza a lista de todos os alvos cadastrados no sistema Tupã.

4.2.4.5 Restrições

1. Não há restrições para este caso de uso.
2. O usuário precisa ser administrador do sistema ou ter a permissão de visualizar alvos.

4.2.5 Mostrar Alvo

Este caso de uso permite a visualização das informações detalhadas de um alvo específico cadastrado no sistema Tupã.

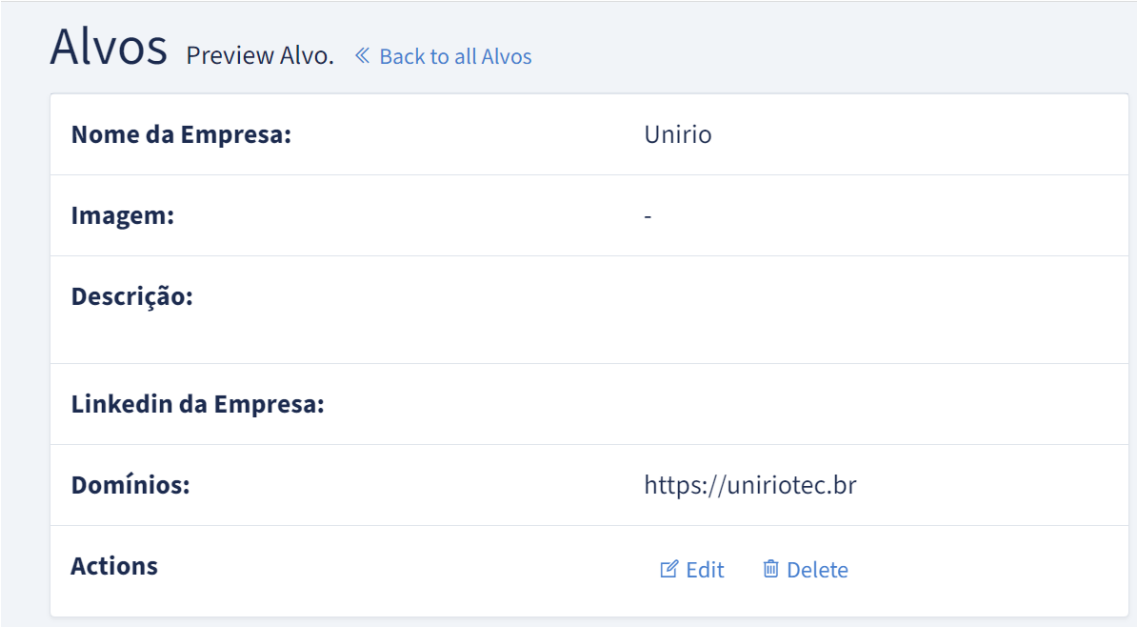
4.2.5.1 Fluxo Básico

1. O usuário escolhe o alvo que deseja visualizar as informações.

Figura 21: Visualização de informações pelo usuário

2. O sistema exibe todas as informações relacionadas ao alvo selecionado.

Figura 22: Visualização de um alvo em específico



4.2.5.2 Pré-condições

1. O usuário deve estar autenticado.
2. Deve haver pelo menos um alvo cadastrado no sistema Tupã.

4.2.5.3 Pós-condições

1. O usuário consegue visualizar todas as informações detalhadas do alvo selecionado.

4.2.5.4 Restrições

1. O usuário não pode editar ou excluir o alvo nesta página, apenas visualizá-lo.
2. O usuário possui a permissão necessária para visualizar o alvo.
3. Caso o alvo não exista no sistema Tupã, o usuário deve ser notificado com uma mensagem de erro.

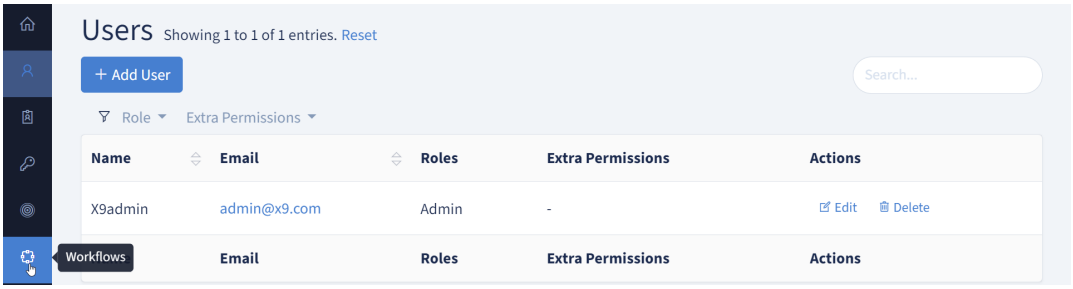
4.2.16 Excluir Workflow

Este caso de uso permite a exclusão de um workflow previamente cadastrado no sistema.

4.2.16.1 Fluxo Básico

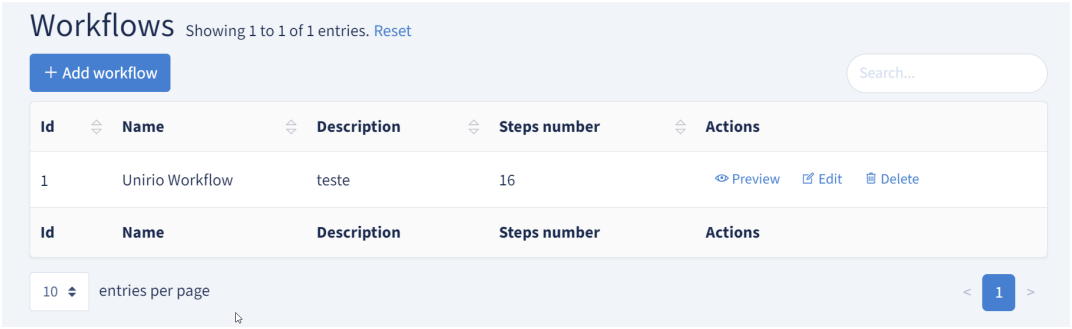
1. O usuário habilitado seleciona a opção workflow no menu.

Figura 23: Selecionar a opção "Workflow" no menu



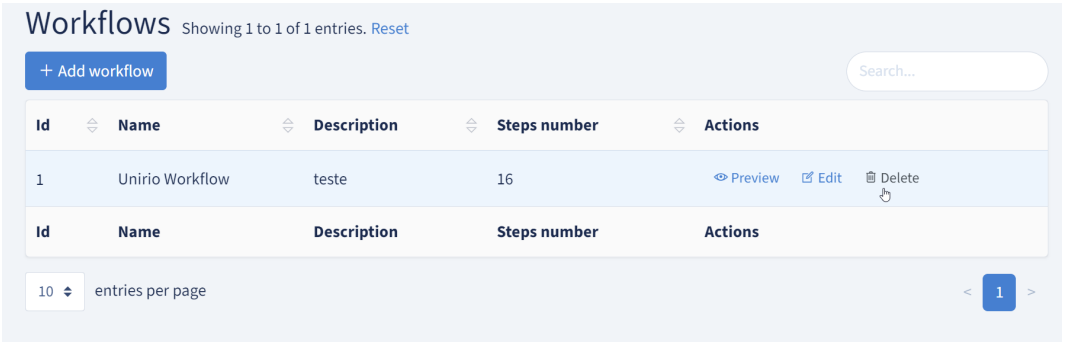
2. O sistema apresenta a lista de workflows cadastrados

.*Figura 24: Lista de Workflows*



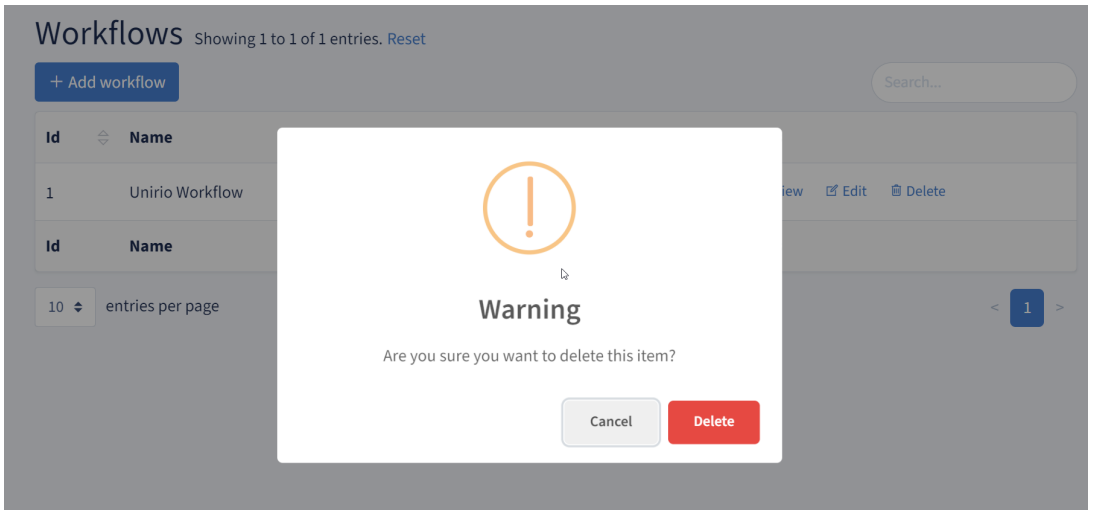
3. O usuário seleciona o workflow que deseja excluir.

Figura 25: Selecionar o workflow que desejar excluir



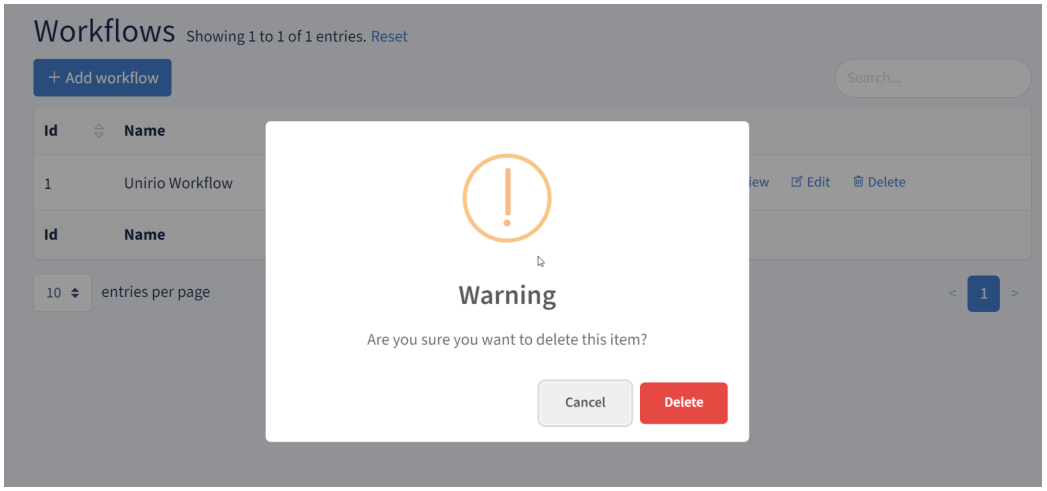
4. O sistema exibe uma tela de confirmação para o usuário.

Figura 26: Exibição de uma tela de confirmação



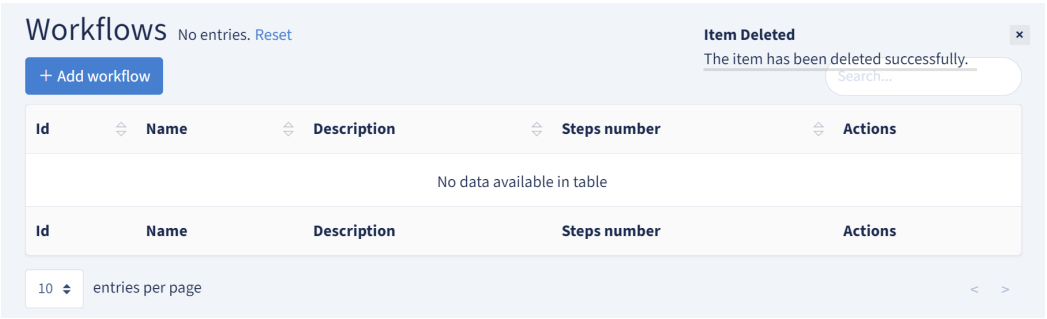
5. O usuário confirma a exclusão.

Figura 27: Confirmar a exclusão



6. O sistema exclui o workflow selecionado do sistema e retorna para a página de listar workflows.
7. O sistema exibe uma mensagem informando que o workflow foi excluído com sucesso.

Figura 28: Exibição do Workflow excluído

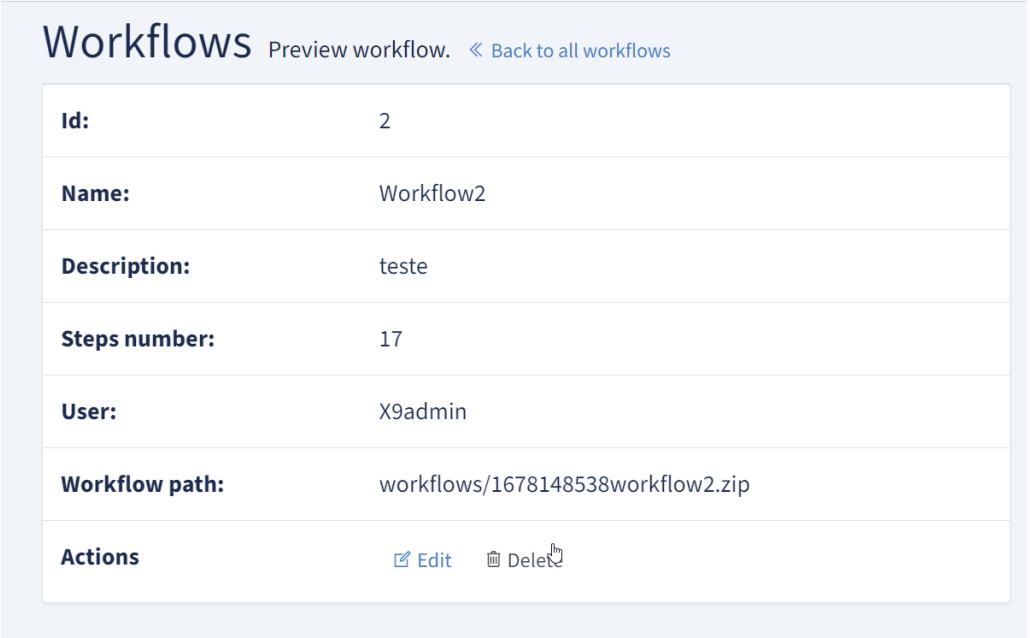


4.2.16.2 Fluxo Alternativo

1a)

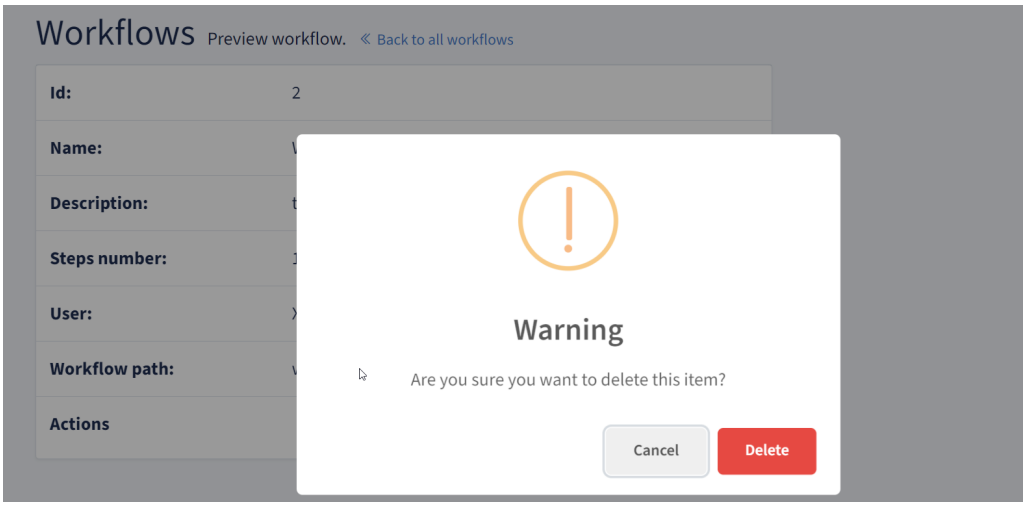
1. A partir da tela visualizar workflow o usuário seleciona o botão excluir.

Figura 29: Selecionar o botão excluir



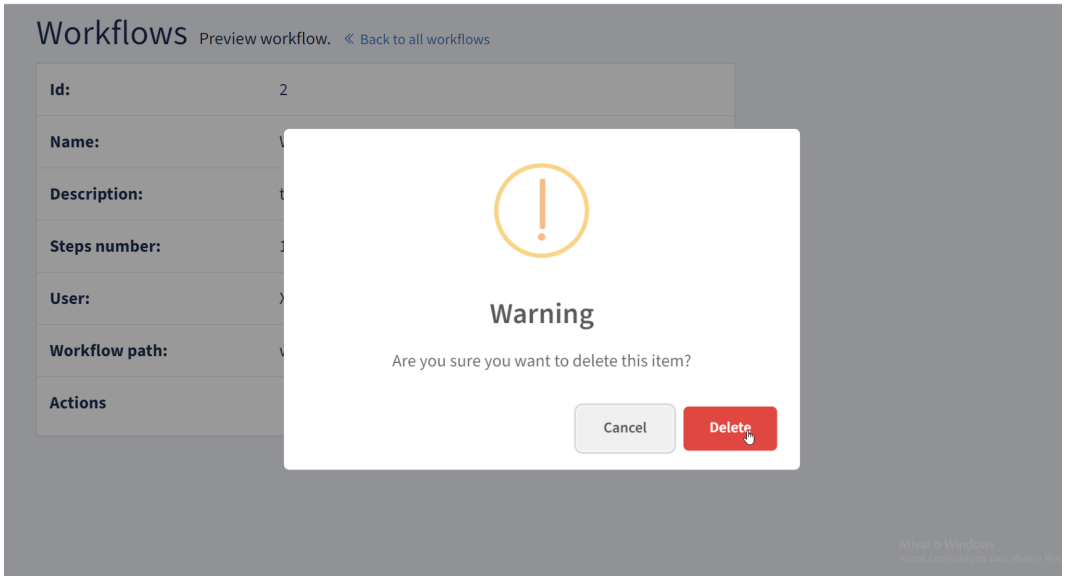
2. O sistema exibe uma tela de confirmação para o usuário.

Figura 30: Exibição de uma tela de confirmação



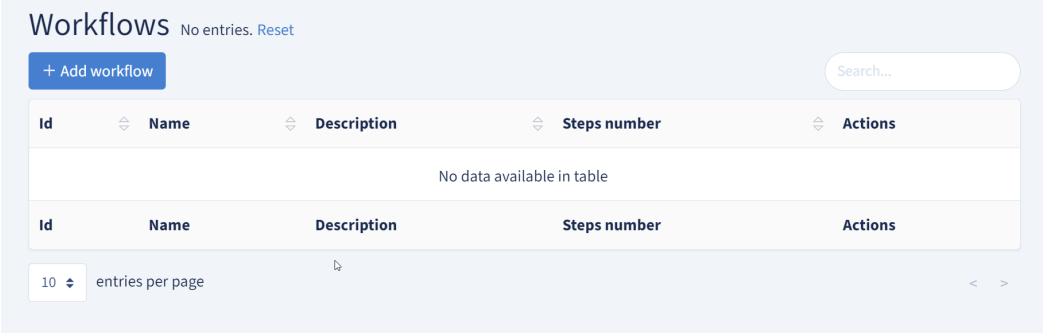
3. O usuário confirma a exclusão.

Figura 31: Confirmar a exclusão



4. O sistema exclui o workflow selecionado do sistema e retorna para a página de listar workflows.

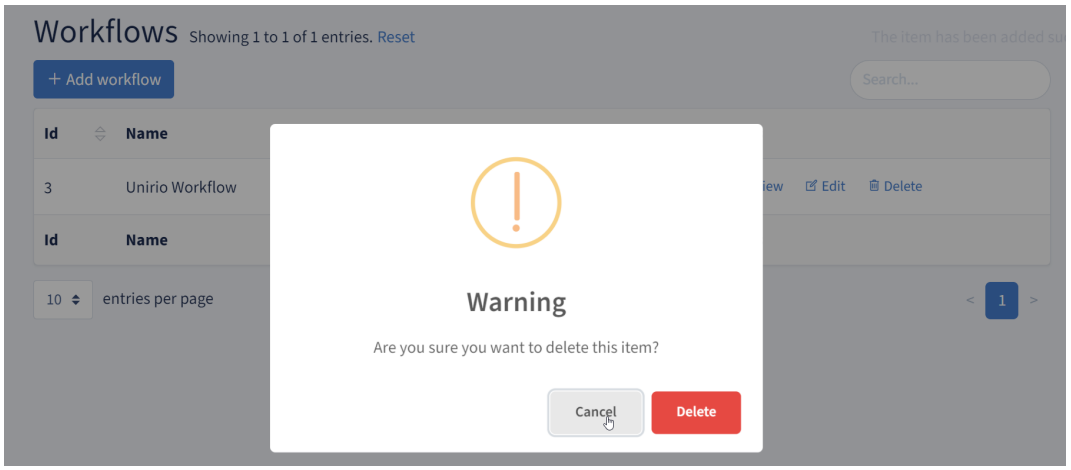
Figura 32: Excluir o workflow e retorna para a página "Listar Workflows"



5a)

1. O usuário desiste da exclusão e cancela a operação, voltando para a tela de listar workflows.

Figura 33: Desistir da exclusão e cancelar operação



5b)

1. O sistema verifica que o workflow possui instâncias em andamento e informa ao usuário que não é possível excluir o workflow neste momento.

4.2.16.2 Pré-condições

1. O usuário precisa estar autenticado no sistema.
2. O usuário precisa estar na tela listar workflow ou visualizar workflow
3. Deve existir pelo menos um workflow cadastrado no sistema.

4.2.16.3 Pós-condições

1. O workflow selecionado é excluído do sistema.
2. O sistema retorna para a página de listar workflows.

4.2.16.4 Restrições

1. Não é possível excluir um workflow que possui instâncias em andamento.
2. O usuário precisa ter a permissão de excluir workflows ou ser um administrador do sistema.

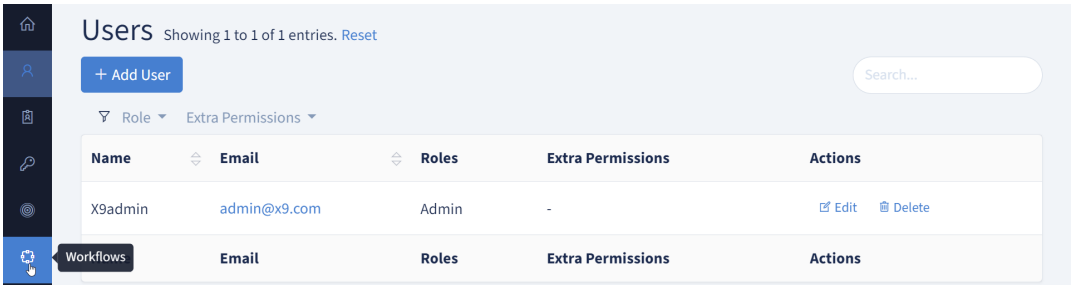
4.2.17 Visualizar Workflow

Esse caso de uso permite a visualização dos detalhes de um workflow específico cadastrado no sistema.

4.2.17.1 Fluxo Básico

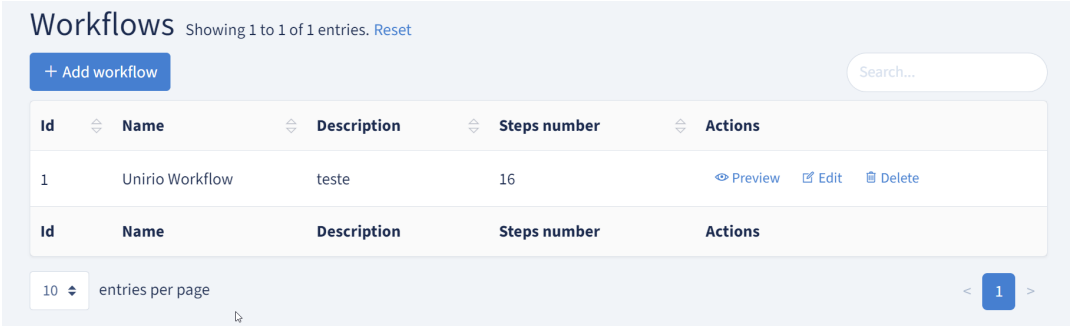
1. O usuário habilitado seleciona a opção “workflows” no menu.

Figura 34: Selecionar opção “Workflows” no menu



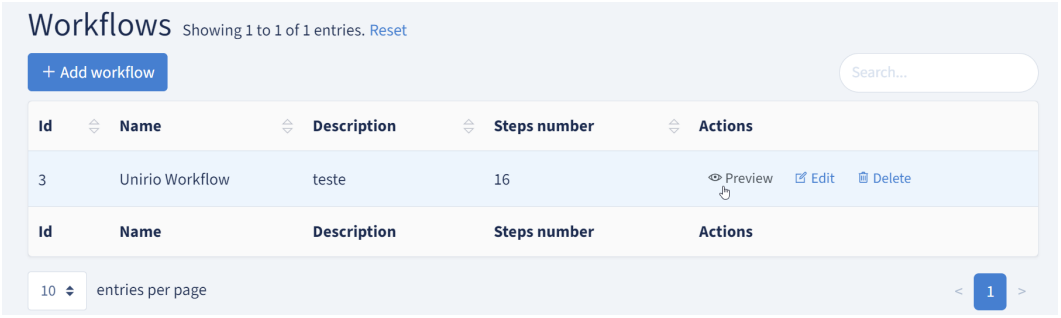
2. O sistema exibe uma lista de workflows cadastrados no sistema.

Figura 35: Lista de Workflows cadastradas no sistema



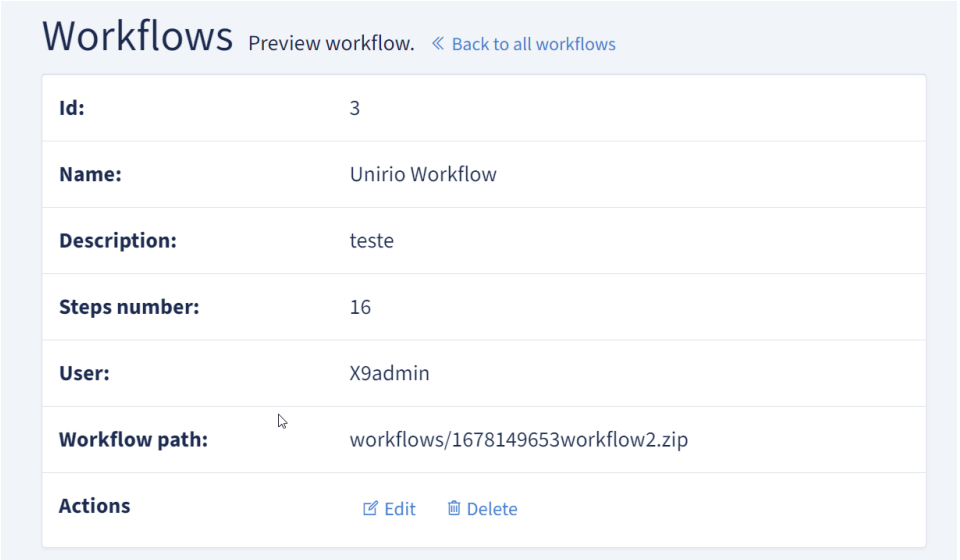
3. O usuário seleciona o workflow desejado.

Figura 36: Selecionar Workflow desejado



4. O sistema exibe as informações detalhadas do workflow, incluindo nome, descrição e quantidade de passos do workflow.

Figura 37: Informações exibidas pelo sistema



4.2.17.2 Fluxo Alternativo

1. Se o usuário não tiver permissão para visualizar workflows, o sistema exibe uma mensagem informando que o usuário não possui permissão para realizar essa ação e volta para a página inicial.

4.2.17.3 Pré-condições

1. O usuário precisa estar autenticado no sistema.
2. O usuário precisa estar na página de listar workflows
3. Deve haver pelo menos um workflow cadastrado no sistema.

4.2.17.4 Pós-condições

1. O usuário visualiza as informações detalhadas do workflow selecionado.

4.2.17.5 Restrições

1. O usuário precisa ter permissão para visualizar workflows ou ser administrador do sistema.

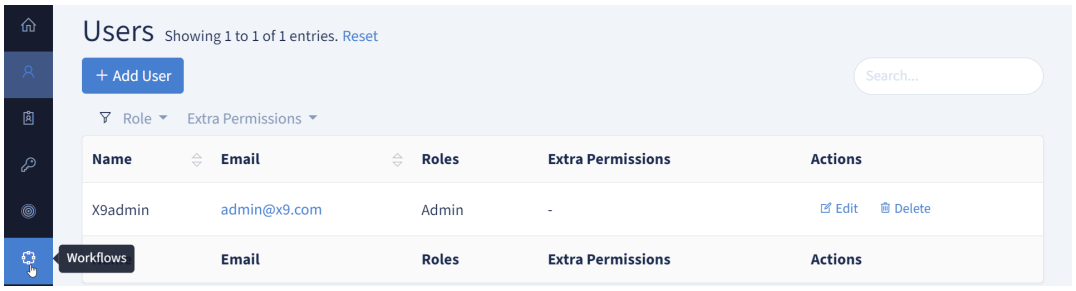
4.2.18 Listar Workflows

Esse caso de uso permite listar todos os workflows cadastrados no sistema.

4.2.18.1 Fluxo Básico

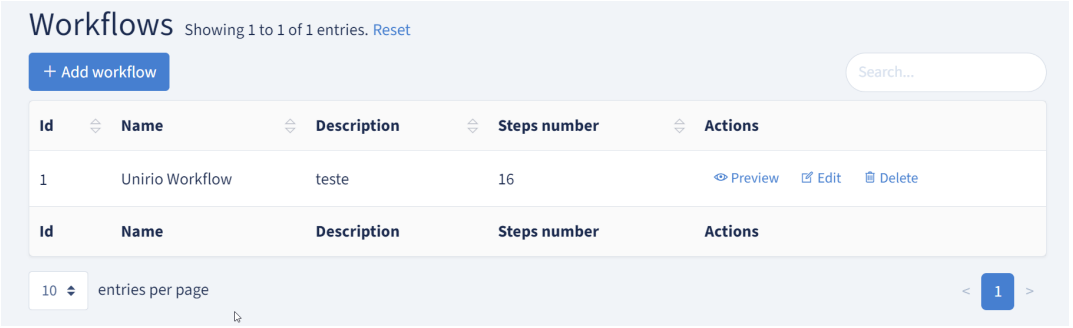
1. O usuário habilitado seleciona a opção "Workflows" no menu.

Figura 38: Selecionar a opção "Workflow" no menu



2. O sistema apresenta uma tabela com todos os workflows cadastrados, mostrando informações como nome e descrição.

Figura 39: O sistema apresenta todas as informações



3. O usuário pode selecionar um workflow específico para visualizar mais informações ou voltar para a página anterior.

4.2.18.2 Pré-condições

1. O usuário precisa estar autenticado no sistema.
2. Existem workflows cadastrados no sistema.

4.2.18.3 Pós-condições

1. O usuário visualiza a lista de todos os workflows cadastrados no sistema.

4.2.18.4 Restrições

1. O usuário precisa ser administrador do sistema ou ter a permissão de visualizar workflows.

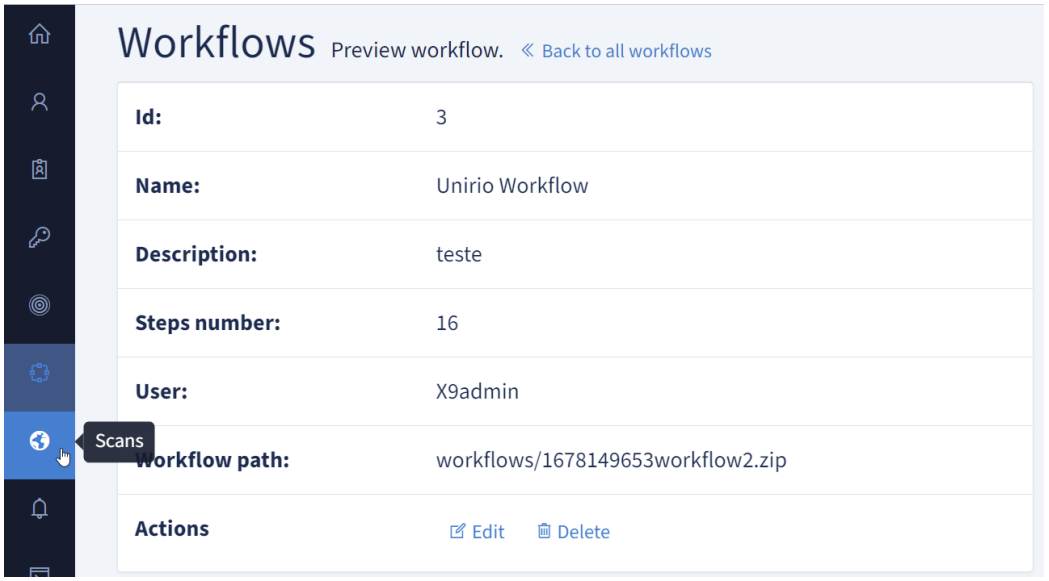
4.2.19 Listar Scans

Este caso de uso permite que o usuário visualize a lista de todos os scans realizados no sistema Tupã.

4.2.19.1 Fluxo Básico

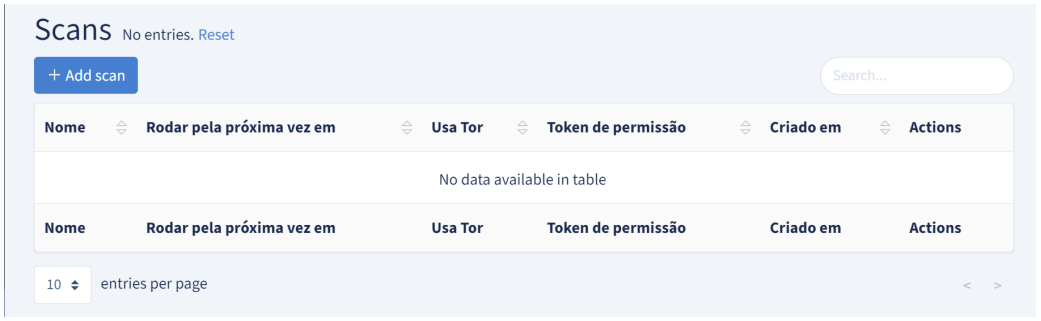
1. O usuário seleciona a opção “Scans” no menu.

Figura 40: Selecionar opção “Scans”



2. O sistema exibe uma tabela contendo todos os scans realizados, mostrando informações como quando o scan roda pela próxima vez, se usa ou não tor, nome e token de permissão.

Figura 41: Tabela contendo todos os scans feitos



3. O usuário pode selecionar um scan específico para visualizar mais informações ou voltar para a página anterior.

4.2.19.2 Pré-condições

1. O usuário precisa estar autenticado no Tupã.
2. Deve haver pelo menos um scan realizado no Tupã.

4.2.19.3 Pós-condições

1. O usuário visualiza a lista de todos os scans realizados no Tupã.

4.2.19.4 Restrições

1. O usuário precisa ser administrador do sistema ou ter a permissão de visualizar scans.

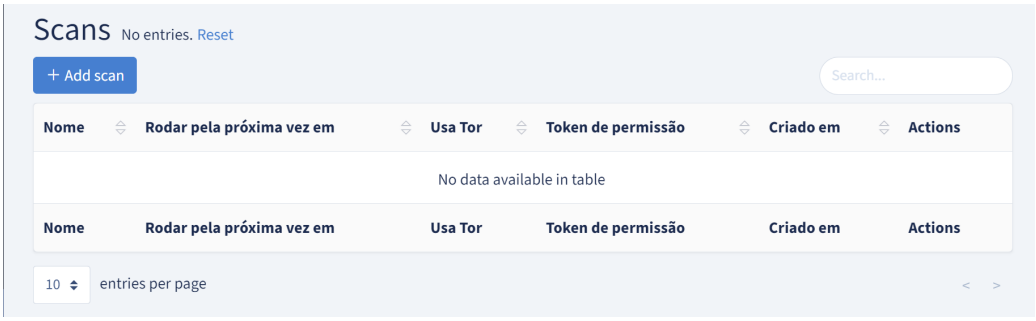
4.2.20 Excluir Scan

Este caso de uso permite a exclusão de um scan previamente realizado no sistema Tupã.

4.2.20.1 Fluxo Básico

1. O usuário acessa a lista de scans realizados no sistema

Figura 42: Acessar a “Lista de Scans”

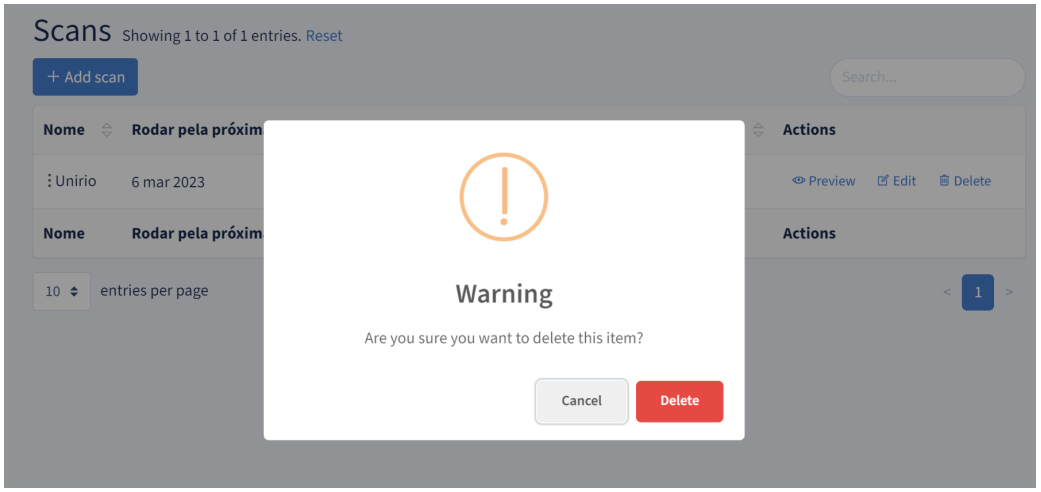


2. O usuário seleciona o scan que deseja excluir.

Figura 43: Selecionar Scan

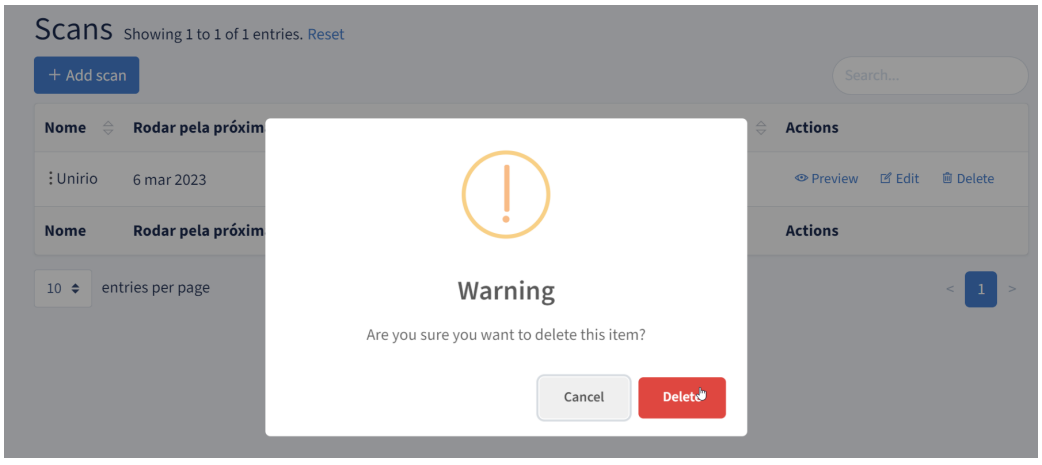
3. O sistema exibe uma mensagem de confirmação para o usuário.

Figura 44: Mensagem de confirmação



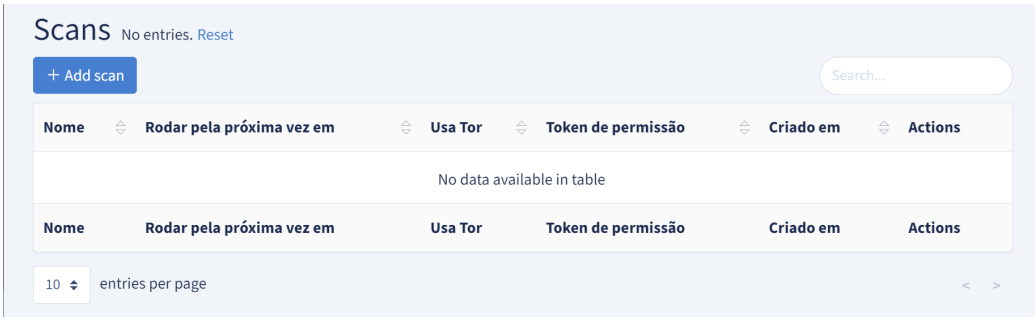
4. O usuário confirma a exclusão do scan.

Figura 45: Confirmar exclusão



5. O sistema exclui o scan selecionado e exibe uma mensagem de sucesso informando que o scan foi excluído com sucesso.

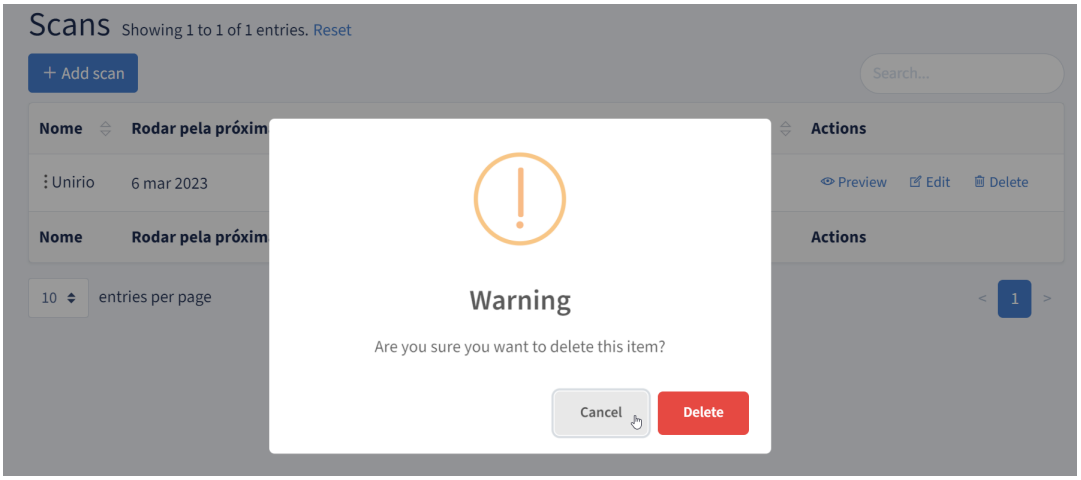
Figura 46: Excluir scan selecionado



4.2.20.2 Fluxo Alternativo

4a. O usuário desiste da exclusão do scan e cancela a operação.

Figura 47: Desistir da exclusão e cancelar operação



4.2.20.3 Pré-condições

1. O usuário precisa estar autenticado no Tupã.
2. Deve haver pelo menos um scan cadastrado no Tupã.

4.2.20.4 Pós-condições

1. O scan selecionado é excluído do sistema.

4.2.20.5 Restrições

1. O usuário precisa ter a permissão para excluir scans ou ser administrador do sistema.

4.2.21 Mostrar Scan

Este caso de uso permite a visualização das informações detalhadas de um scan específico realizado em um alvo previamente cadastrado no sistema Tupã.

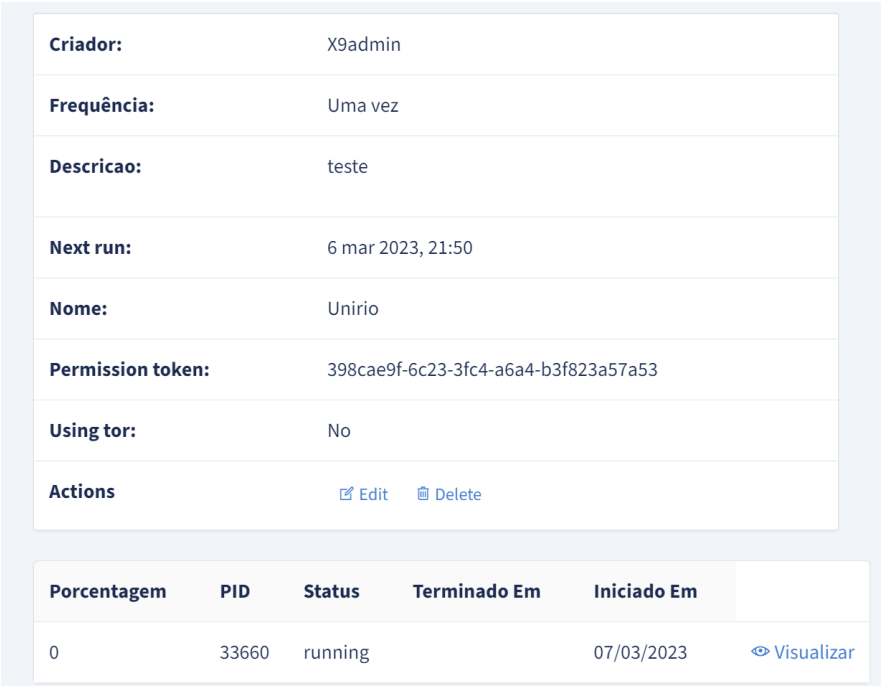
4.2.21.1 Fluxo Básico

1. Em listar scans o usuário seleciona o scan que deseja visualizar as informações.

Figura 48: O usuário seleciona no scan que desejar visualizar em " Listar Scans"

2. O sistema exibe todas as informações relacionadas ao scan selecionado, incluindo suas informações e informações relacionadas às suas tarefas.

Figura 49: O sistema exibe todos os dados



4.2.21.2 Pré-condições

1. O usuário deve estar autenticado.
2. Deve haver pelo menos um scan realizado para o alvo selecionado.

4.2.21.3 Pós-condições

1. O usuário consegue visualizar todas as informações detalhadas do scan selecionado.

4.2.21.4 Restrições

1. O usuário não pode editar ou excluir o scan nesta página, apenas visualizá-lo.
2. O usuário possui a permissão necessária para visualizar o scan.
3. Caso o scan não exista no sistema Tupã, o usuário deve ser notificado com uma mensagem de erro.

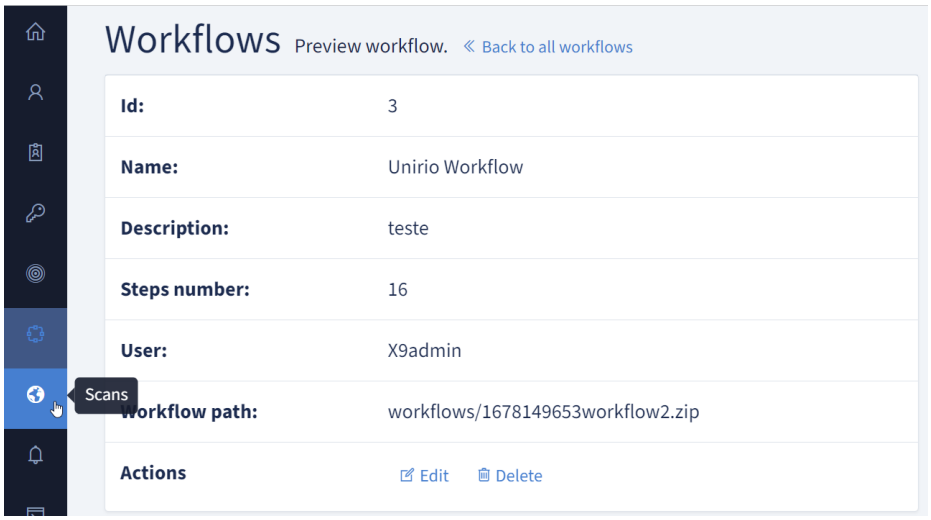
4.2.22 Visualizar Resultado do Scan

Esse caso de uso permite ao usuário visualizar os resultados de um scan previamente realizado no sistema Tupã.

4.2.22.1 Fluxo Básico

1. O usuário seleciona “scans” no menu.

Figura 50: Selecionar “ Scans” no menu

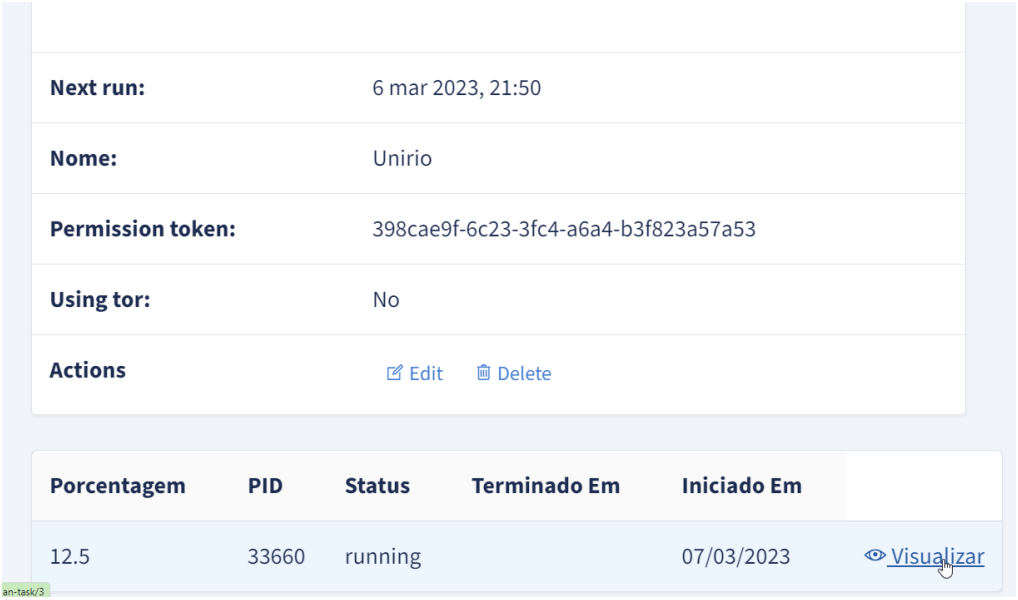


2. O usuário seleciona o scan do qual deseja visualizar os resultados.

Figura 51: Selecionar Scan

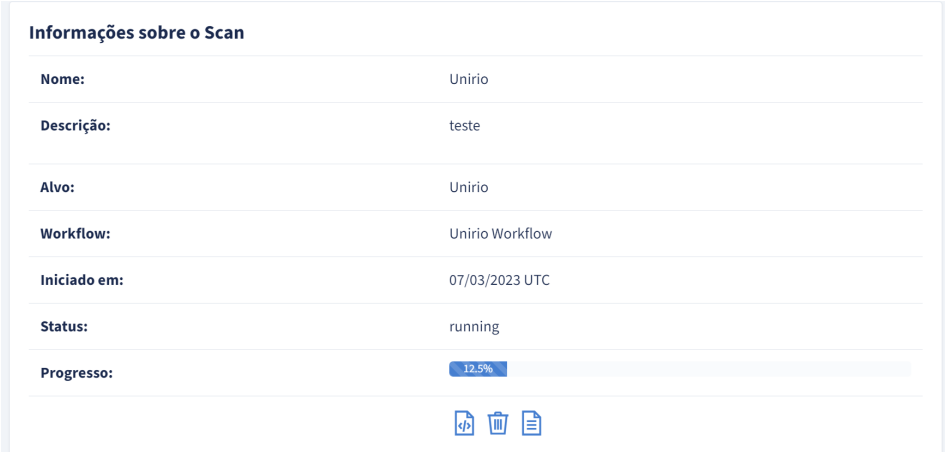
3. O usuário seleciona uma tarefa de scan executada.

Figura 52: Selecionar tarefa de Scan executada



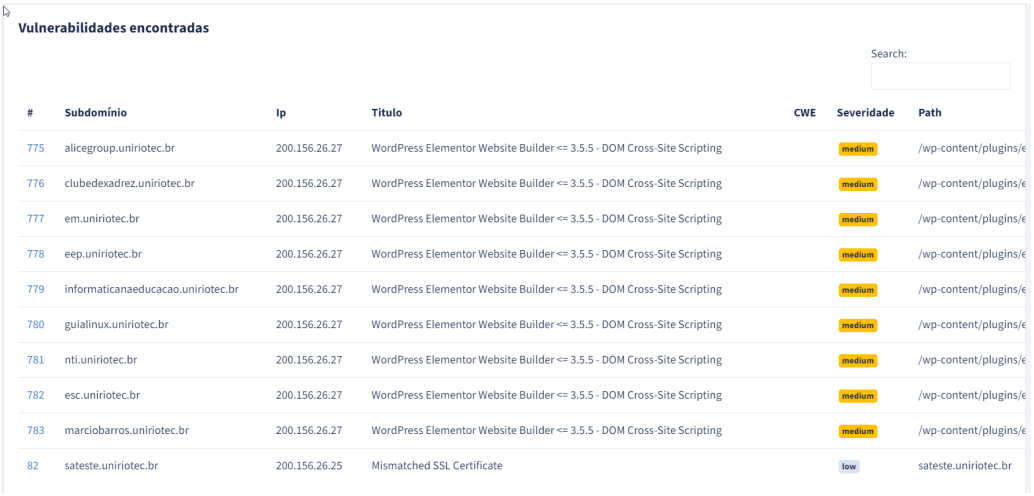
4. O sistema exibe uma tela com um resumo dos resultados do scan, incluindo as vulnerabilidades encontradas, informações de hosts e outras informações relevantes.

Figura 53: Sistema exibe uma tabela com resultados



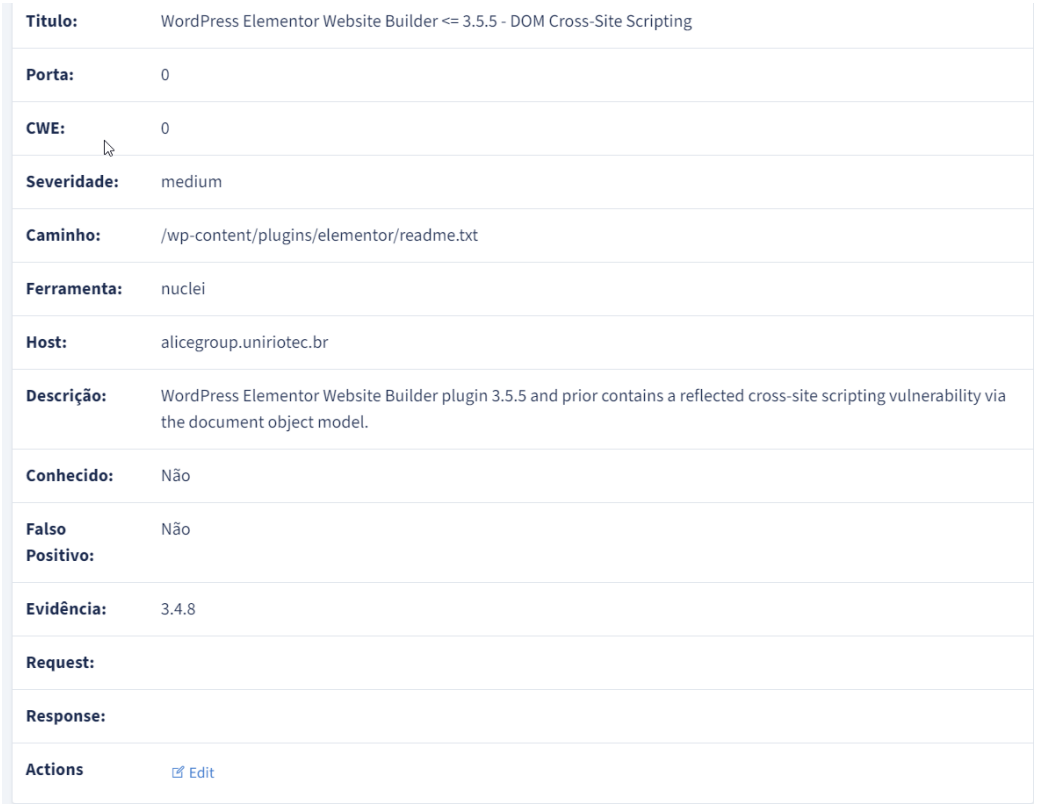
5. O usuário pode selecionar uma vulnerabilidade específica para visualizar mais informações sobre ela.

Figura 54: Selecionar uma vulnerabilidade específica para saber mais dados sobre ela



6. O sistema exibe mais informações sobre a vulnerabilidade selecionada, incluindo detalhes sobre o host afetado, a gravidade da vulnerabilidade e possíveis soluções.

Figura 55: Mais informações sobre a vulnerabilidade selecionada



Titulo:	WordPress Elementor Website Builder <= 3.5.5 - DOM Cross-Site Scripting
Porta:	0
CWE:	0
Severidade:	medium
Caminho:	/wp-content/plugins/elementor/readme.txt
Ferramenta:	nuclei
Host:	alicegroup.uniriotec.br
Descrição:	WordPress Elementor Website Builder plugin 3.5.5 and prior contains a reflected cross-site scripting vulnerability via the document object model.
Conhecido:	Não
Falso Positivo:	Não
Evidência:	3.4.8
Request:	
Response:	
Actions	Edit

7. O usuário pode voltar à tela de resumo do scan.

4.2.22.2 Fluxos Alternativos

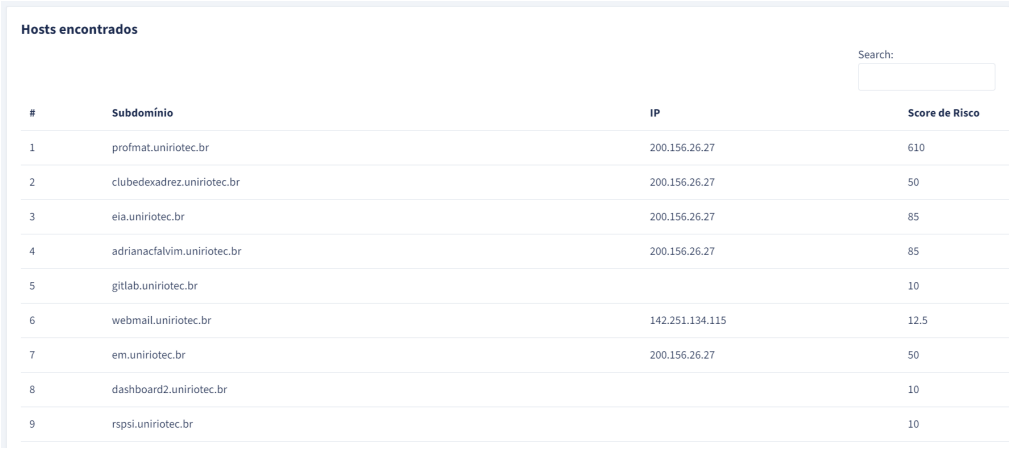
2a)

1. Se o usuário não tiver permissão para visualizar os resultados do scan, o sistema exibe uma mensagem de erro informando que o usuário não tem autorização para acessar essa página.

5a)

1. O usuário pode selecionar um subdomínio descoberto para ter acesso a informações sobre ele, como vulnerabilidades específicas, portas e endereços de rede.

Figura 56: Selecionar domínio para obter mais informações.



4.2.22.3 Pré-condições

1. O usuário precisa estar autenticado no sistema Tupã.
2. Deve haver pelo menos um scan realizado no sistema.

4.2.22.4 Pós-condições

1. O usuário pode visualizar os resultados do scan selecionado.

4.2.22.5 Restrições

1. O usuário deve ter a permissão necessária para visualizar os resultados do scan ou ser um administrador do sistema.

4.2.23 Criar Scan

Este caso de uso permite a criação de um novo scan para um alvo específico.

4.2.23.1 Pré-condições

1. O usuário deve estar autenticado no sistema Tupã.
2. Deve existir pelo menos um alvo cadastrado no sistema.

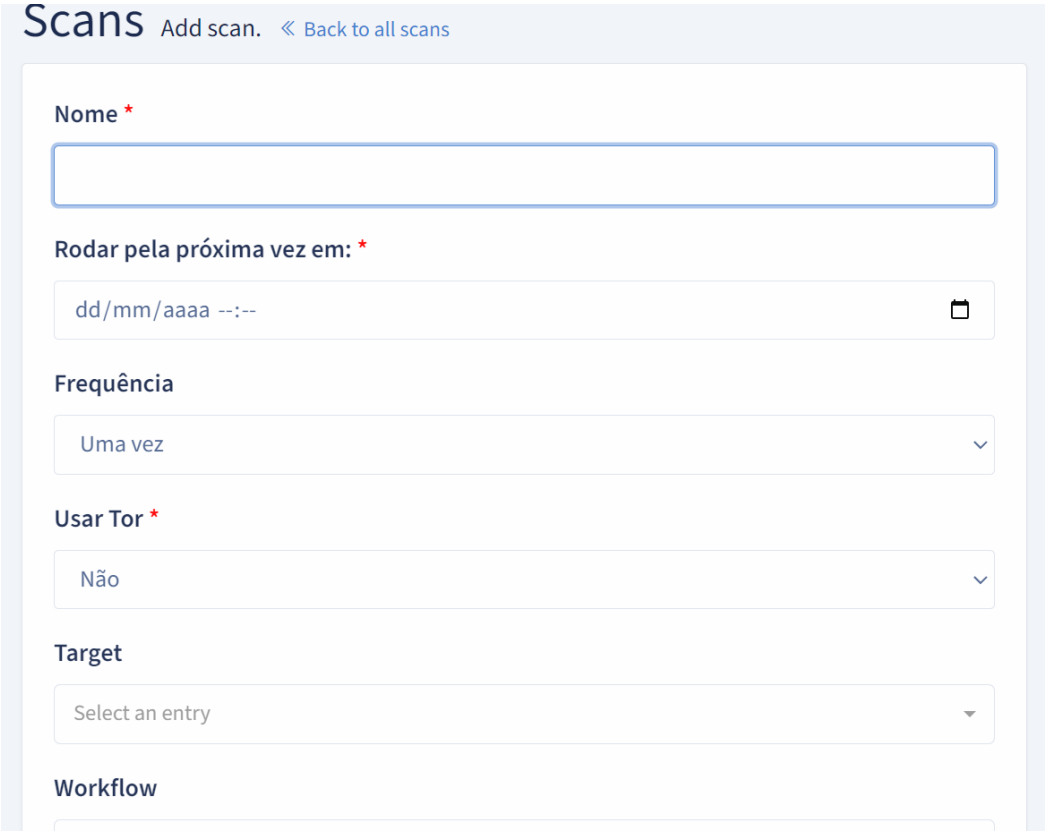
4.2.23.2 Fluxo Básico

1. O usuário habilitado seleciona a opção "Criar Scan" na página de “Listar Scans”

Figura 57: Selecionar a opção “Criar Scan” na página de “Listar Scans”

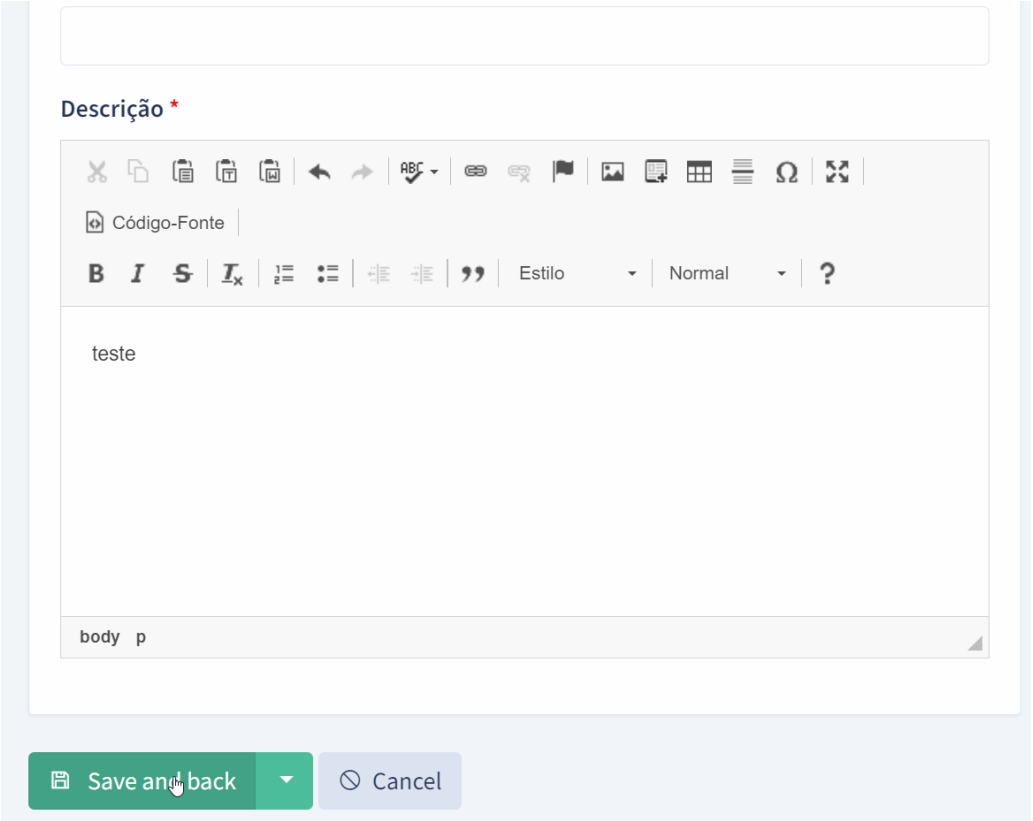
2. O sistema apresenta as opções de configuração para o novo scan, como o workflow selecionado, nome, descrição, alvo, frequência, horário, se possui alertas.

Figura 58: Opções de configurações para o novo Scan



3. O usuário configura as opções do scan de acordo com as suas necessidades.
4. O usuário confirma as opções de configuração do scan.

Figura 59: Confirmar opções de configuração do novo Scan

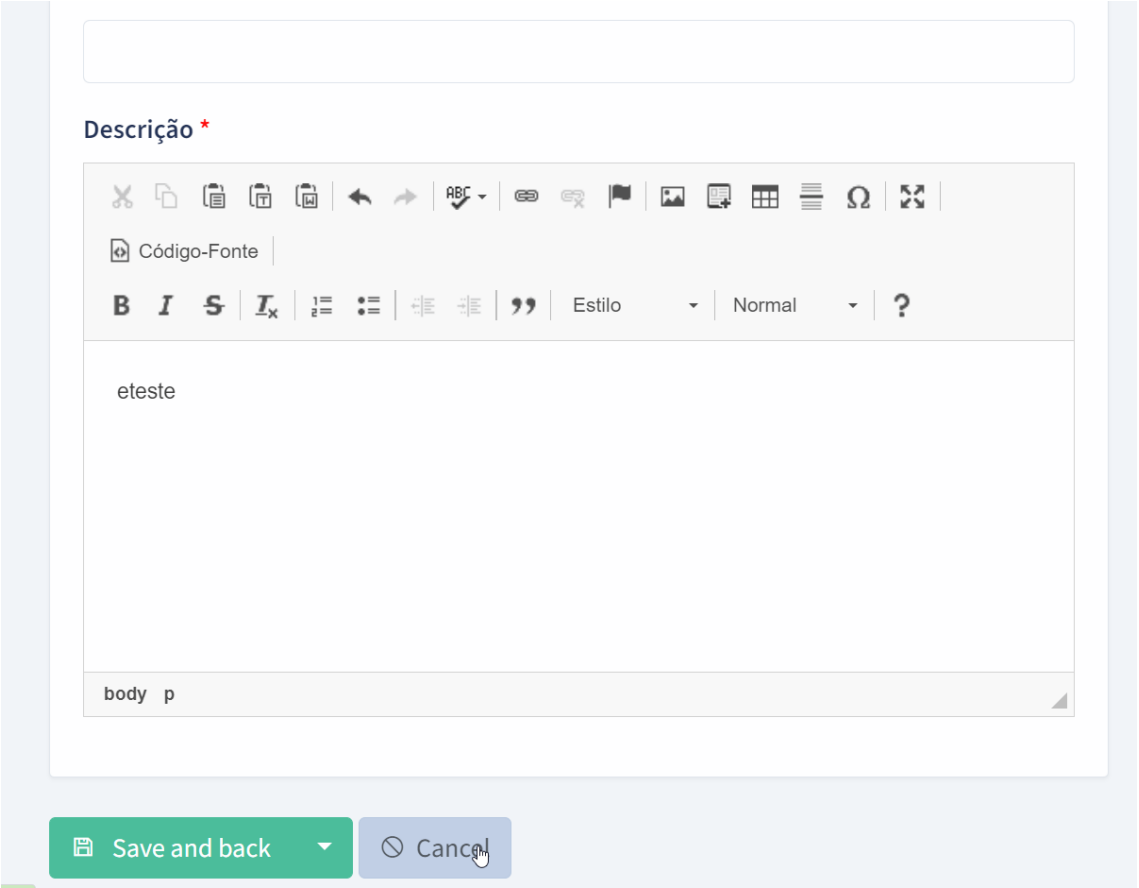


5. O sistema cria um novo scan com as opções de configuração escolhidas e agenda a execução do scan.
6. O sistema exibe uma mensagem de sucesso informando que o scan foi criado e inicia a exibição de tarefas do scan.

4.2.23.3 Fluxos Alternativos

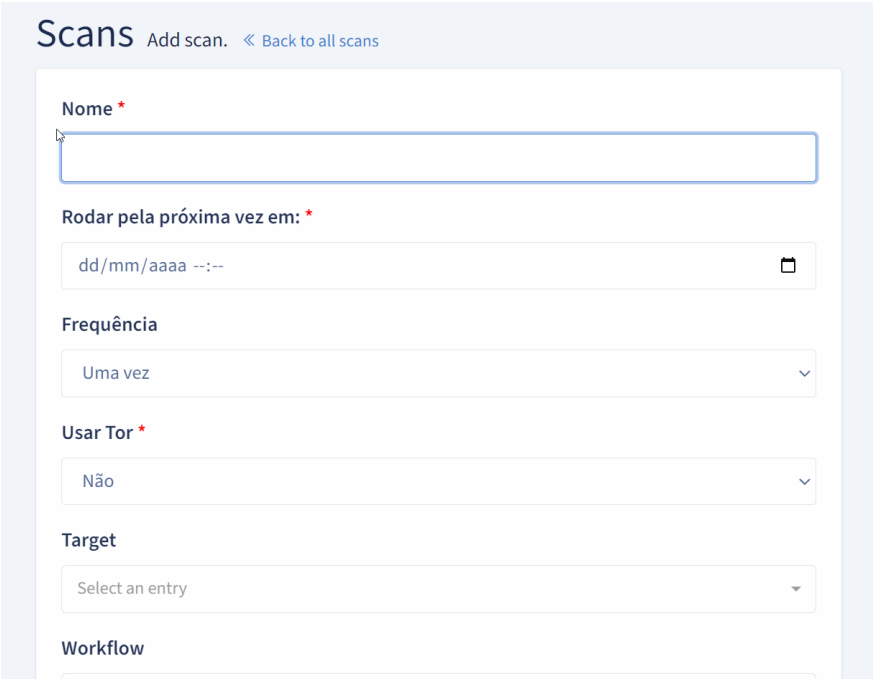
3a. O usuário desiste de criar o scan e cancela a operação.

Figura 60: Desistir de criar Scan e cancelar operação



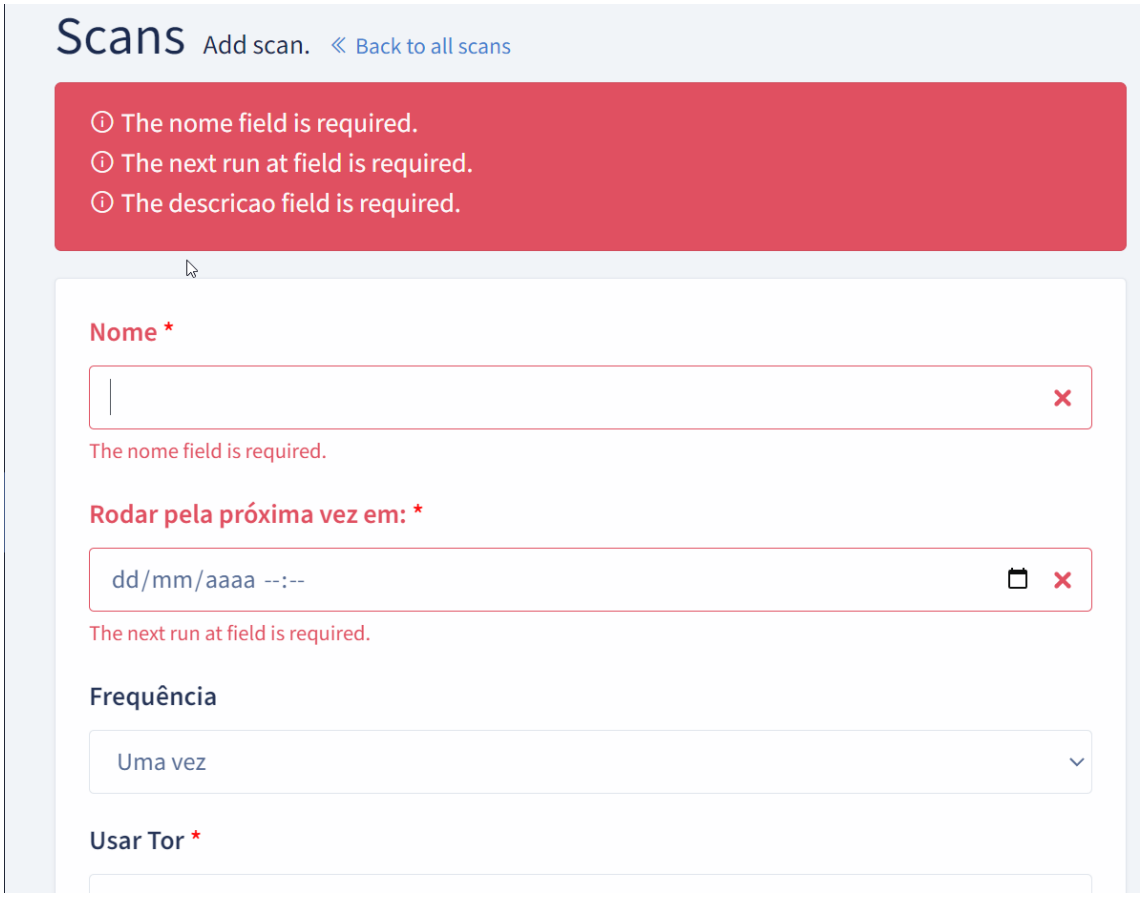
3b. O usuário escolhe opções inválidas para o scan.

Figura 61: Scan com opções inválidas.



1. O sistema emite uma mensagem de erro informando que as opções escolhidas são inválidas e solicita ao usuário que corrija as opções.

Figura 62: Opções escolhidas são inválidas e solicitação de correção.



2. O usuário habilitado realiza a correção das opções e confirma novamente as opções de configuração do scan.

4.2.23.3 Pós-condições

1. Um novo scan é adicionado ao sistema.
2. O sistema inicia a execução do novo scan.

4.2.23.4 Restrições

1. O usuário precisa ter a permissão para criar scans ou ser administrador do sistema.
2. O alvo selecionado precisa existir no sistema Tupã.
3. O usuário habilitado só pode criar um scan para um alvo específico de cada vez.

4.2.24 Editar Scan

Este caso de uso permite a edição das informações de um scan previamente cadastrado no sistema Tupã.

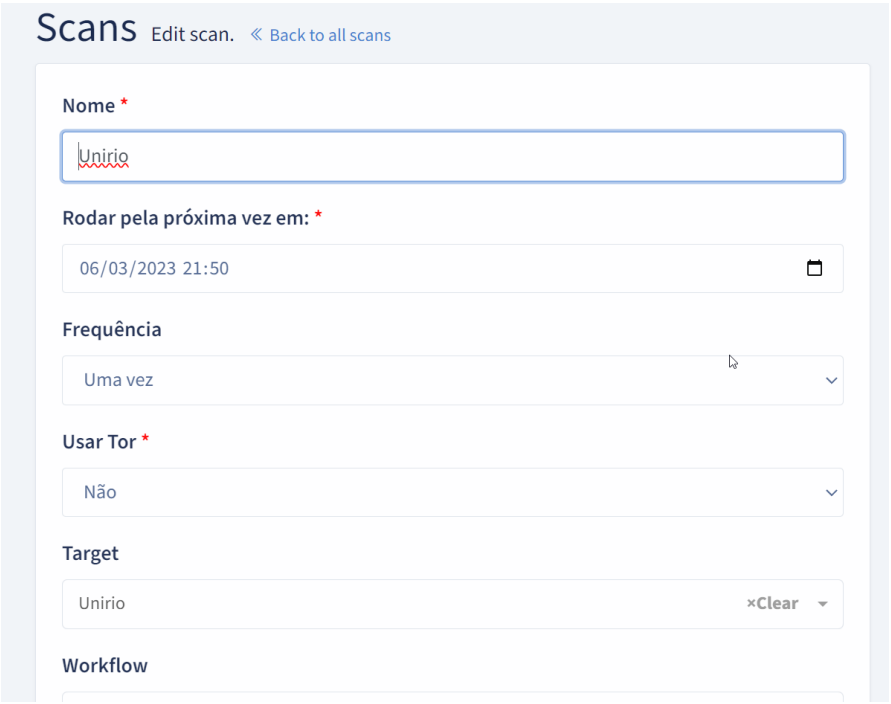
4.2.24.1 Fluxo Básico

1. O usuário habilitado seleciona a opção de editar scan na página onde os scans são listados.

Figura 63: Selecionar a opção de editar Scan na página onde os scans são listados

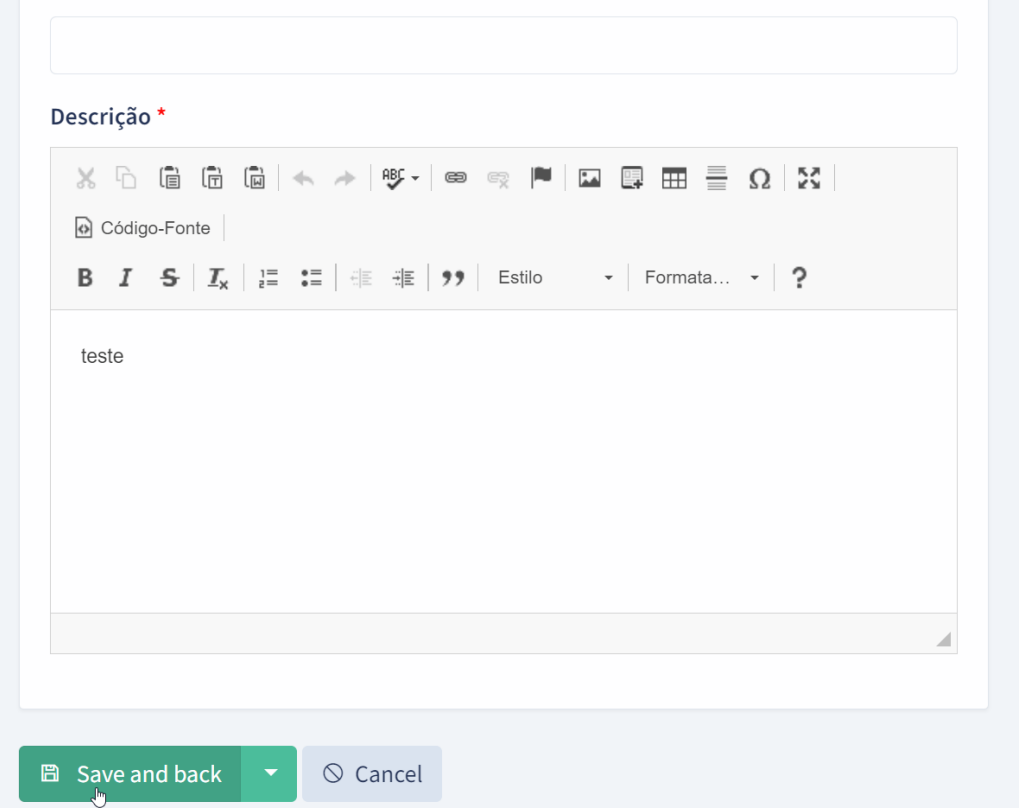
2. O sistema exibe as informações do scan selecionado.

Figura 64: Informações selecionadas do Scan



3. O usuário modifica as informações do scan, como alvo, workflow, agendamento e alertas.
4. O usuário confirma a alteração realizada.

Figura 65: Confirmação da alteração realizada



5. O sistema atualiza as informações do scan.
6. O usuário é redirecionado para a página de listar scans.

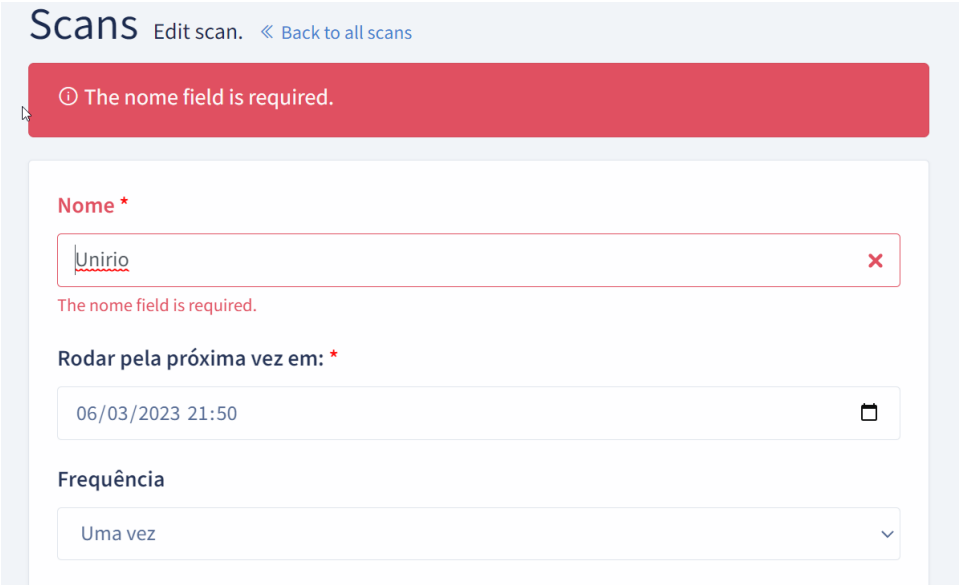
Figura 66: Redirecionamento para a página "Listar Scans"

4.2.24.2 Fluxo Alternativo

1. O usuário modifica as informações do scan com dados inválidos.

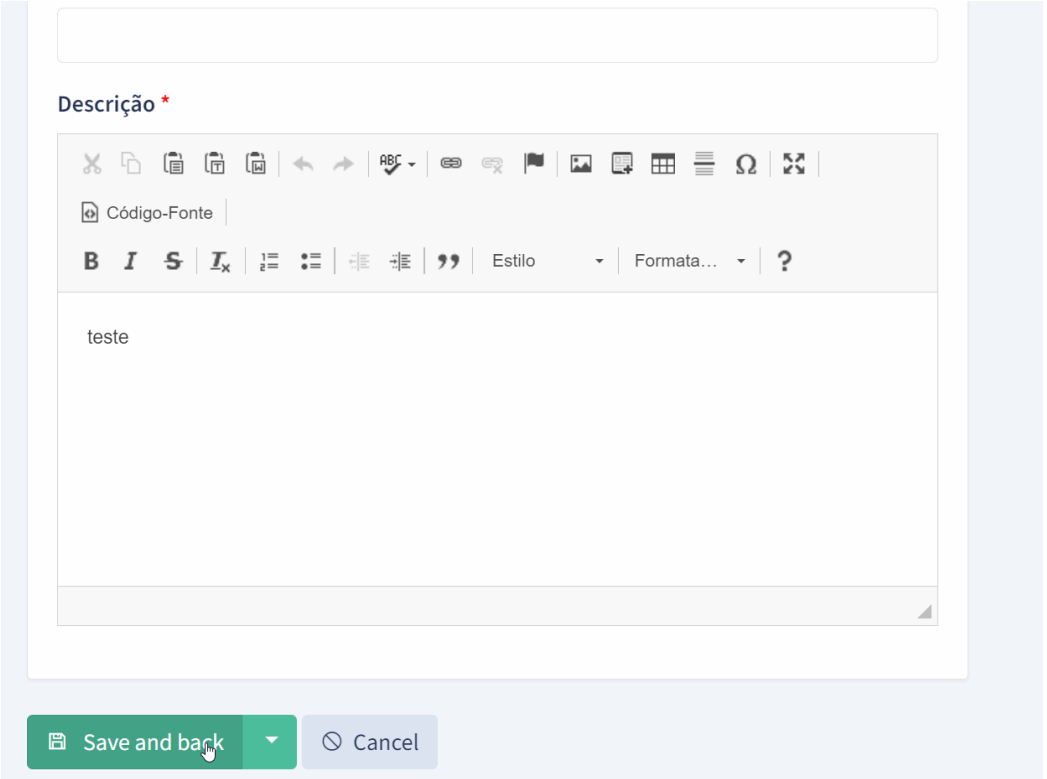
2. O sistema emite uma mensagem de erro informando que os dados são inválidos e solicita ao usuário que insira informações válidas.

Figura 67: Mensagem indicando erro



3. O usuário realiza a correção dos dados e confirma a alteração novamente.

Figura 68: Correção das informações e confirmação da alteração



4.2.24.3 Pré-condições

1. O usuário deve estar autenticado.
2. O scan selecionado deve existir no sistema.

4.2.24.4 Pós-condições

1. As informações do scan selecionado são atualizadas no sistema.

4.2.24.5 Restrições

1. O scan deve ter um alvo válido associado.
2. O scan deve ter um workflow associado.
3. O scan deve ter uma data de ínicio associada
4. O usuário deve possuir a permissão de modificar scans ou ser administrador do sistema.

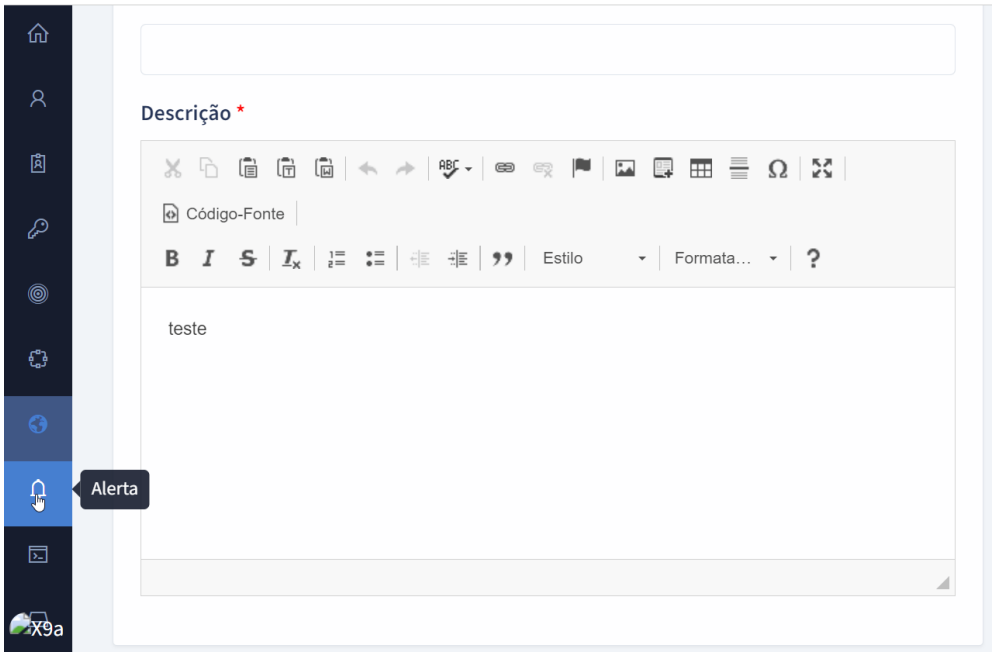
4.2.25 Criar Alerta

Este caso de uso permite a criação de um alerta para um determinado alvo.

4.2.25.1 Fluxo Básico

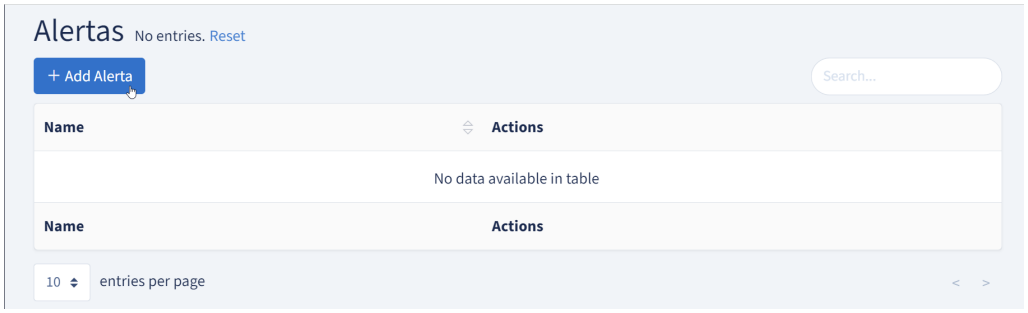
1. O usuário habilitado seleciona a opção "alerta" no menu.

Figura 69: Selecionar a opção "Alerta"



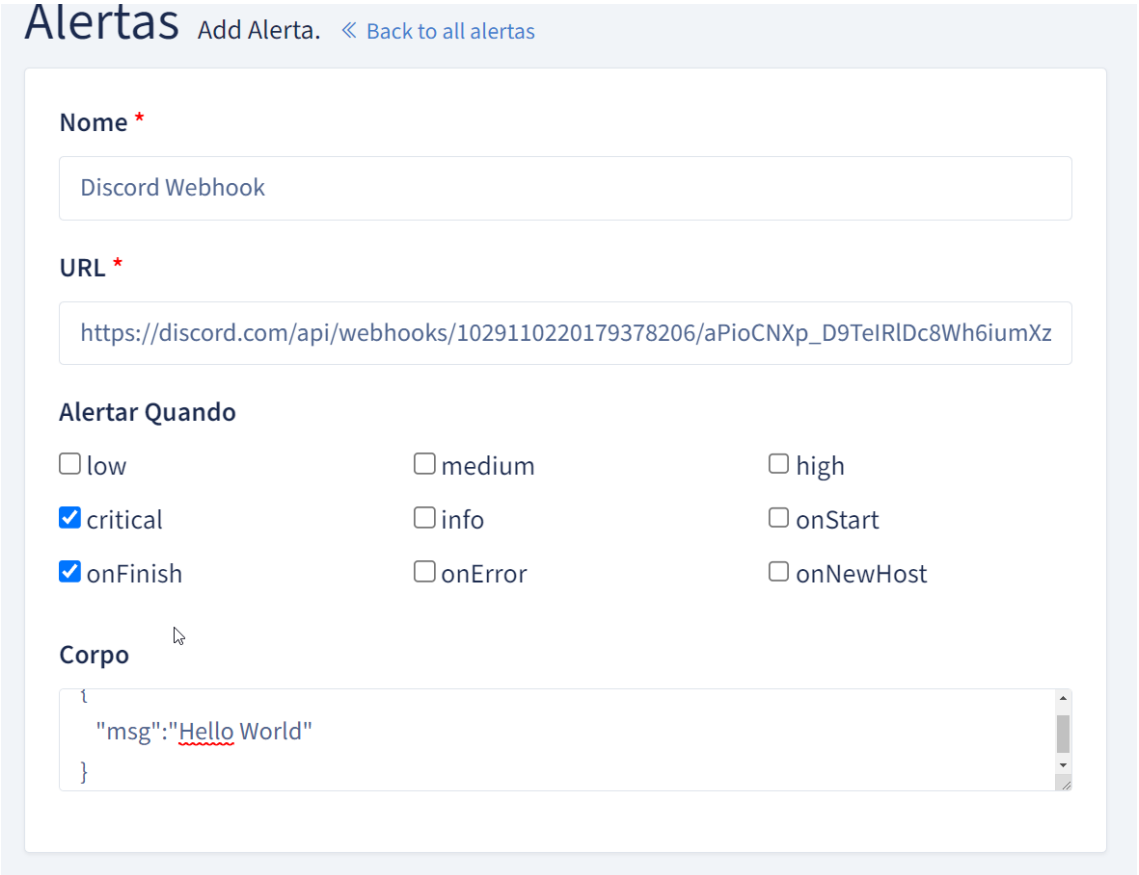
2. Em listar alertas o usuário seleciona a opção "adicionar alerta"

Figura 70: Selecionar a opção "Adicionar Alerta"



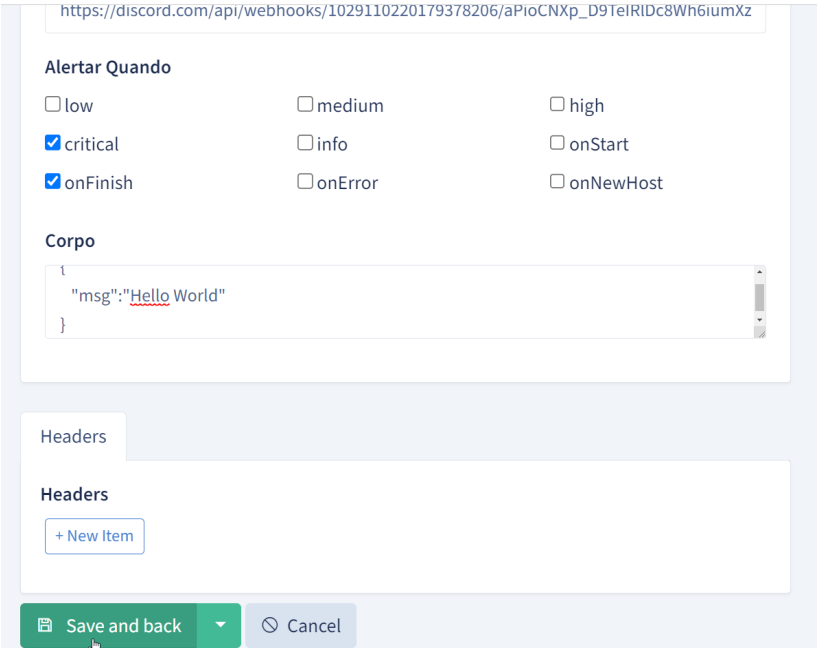
3. O usuário habilitado informa o nome do alerta, a URL para onde enviar o alerta, descrição, corpo do alerta em JSON e eventuais cabeçalhos Http necessários.

Figura 71: Informações são fornecidas



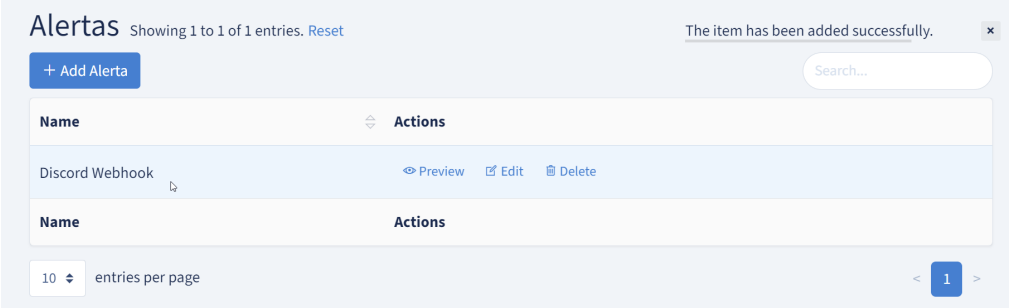
4. O usuário confirma as informações fornecidas.

Figura 72: Confirmar informações fornecidas



5. O sistema salva o novo alerta e retorna para a página do alvo.

Figura 73: Salvar novo alerta e retornar para a página do alvo

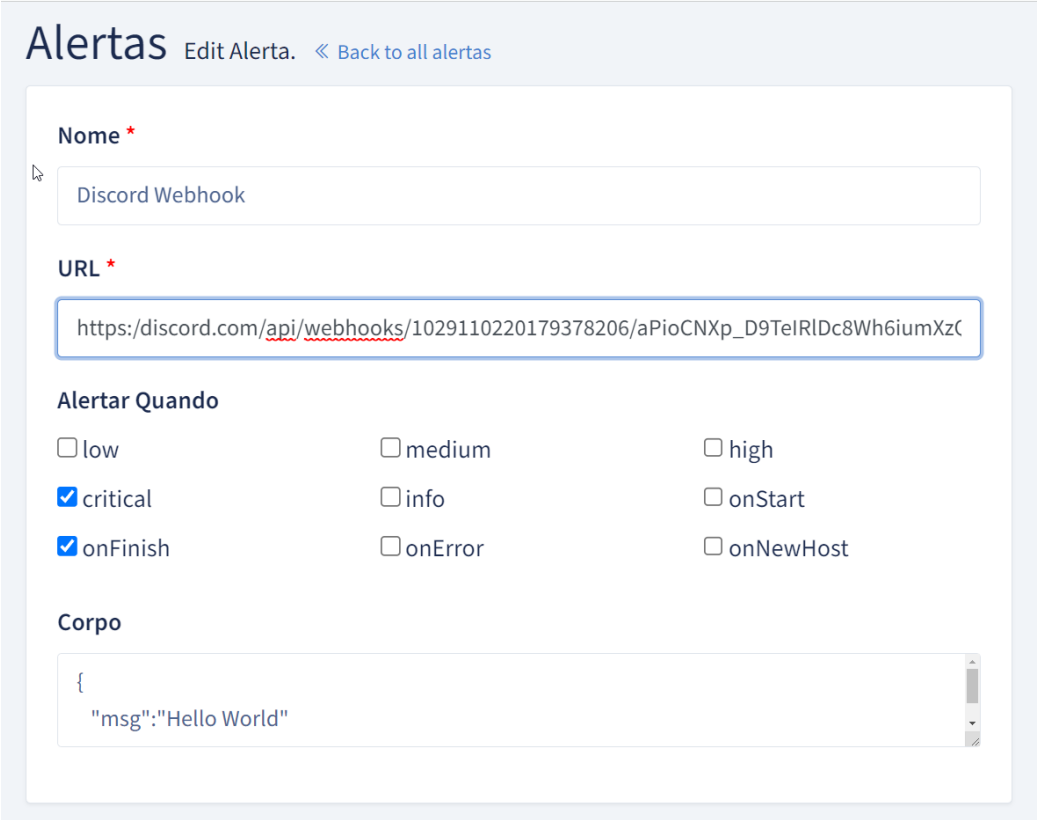


4.2.25.2 Fluxo Alternativo

2a)

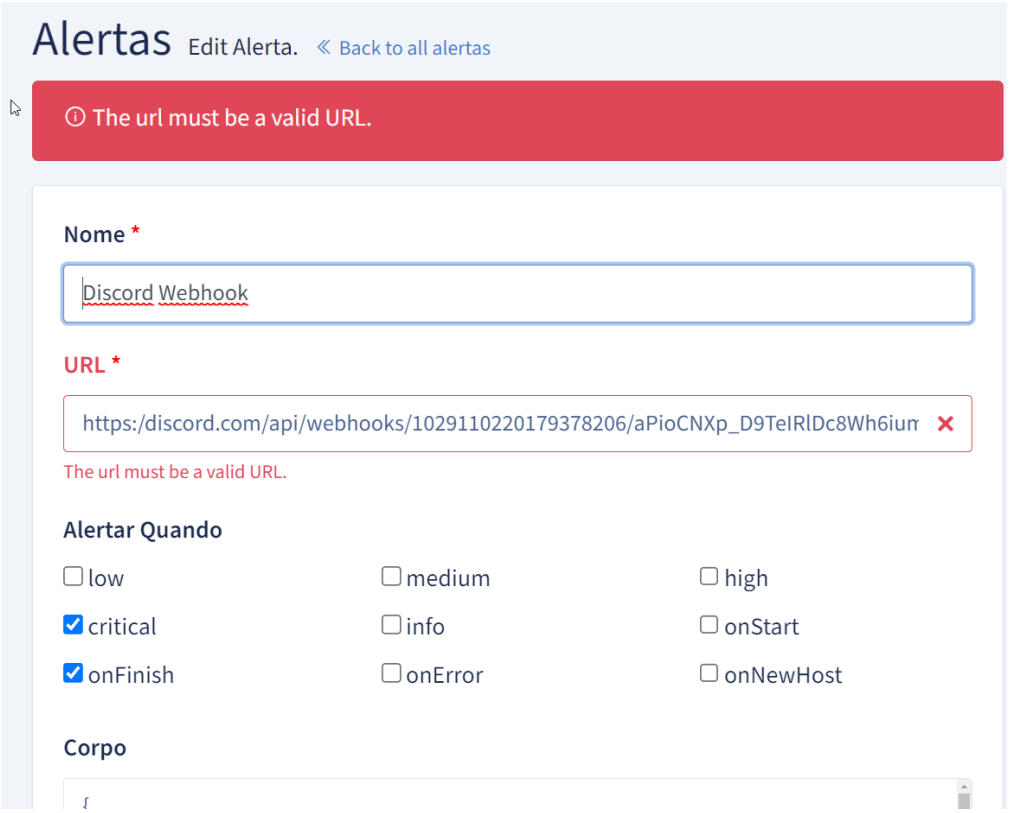
1. O usuário informa uma URL inválida.

Figura 74: URL inválida



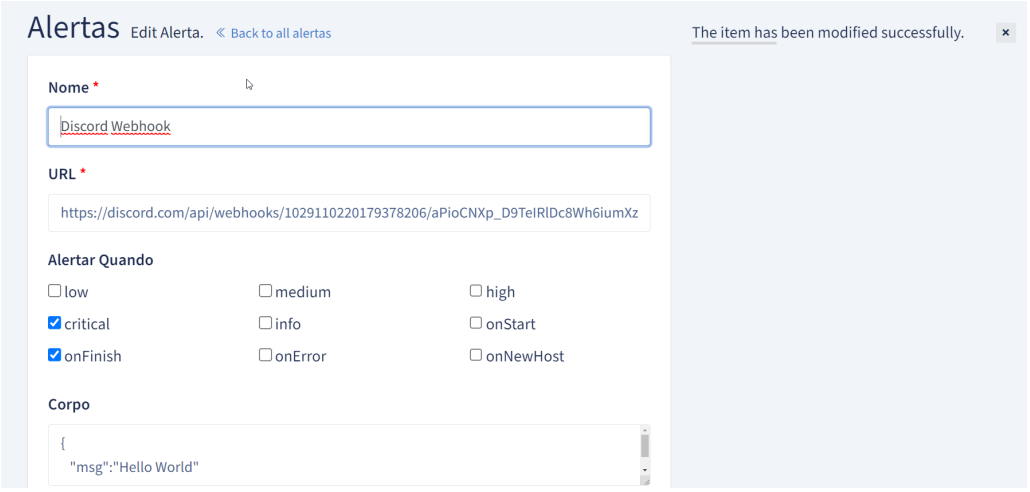
2. O sistema emite uma mensagem de erro informando que a URL é inválida e solicita que ele insira uma URL válida.

Figura 75: Mensagem de erro informando que a URL é inválida



3. O usuário corrige a informação e confirma novamente.

Figura 76: Corrigir a informação e confirmar.



4.2.25.3 Pré-condições

1. O usuário precisa estar autenticado no sistema Tupã.

4.2.25.4 Pós-condições

1. Um novo alerta é criado para ser adicionado a um ou mais scanners no Tupã.

4.2.25.5 Restrições

1. O nome do alerta deve ter pelo menos 5 caracteres.
2. A URL para onde enviar o alerta deve ser uma URL válida
3. O usuário precisa ter permissão para criar alertas.

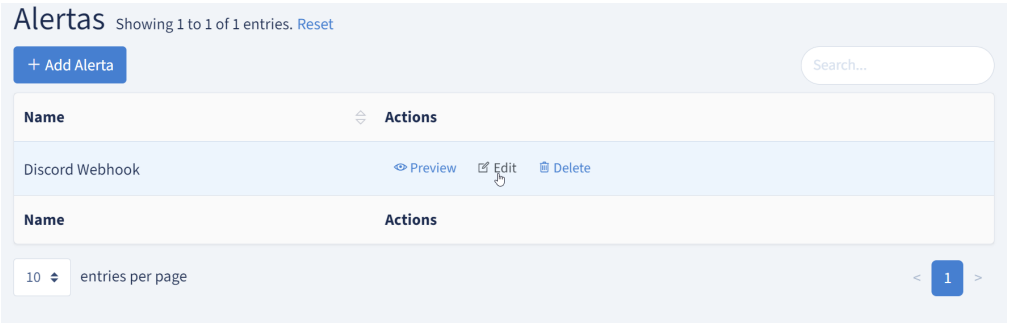
4.2.26 Editar Alerta

Esse caso de uso permite a edição das informações de um alerta previamente cadastrado no sistema Tupã.

4.2.26.1 Fluxo Básico

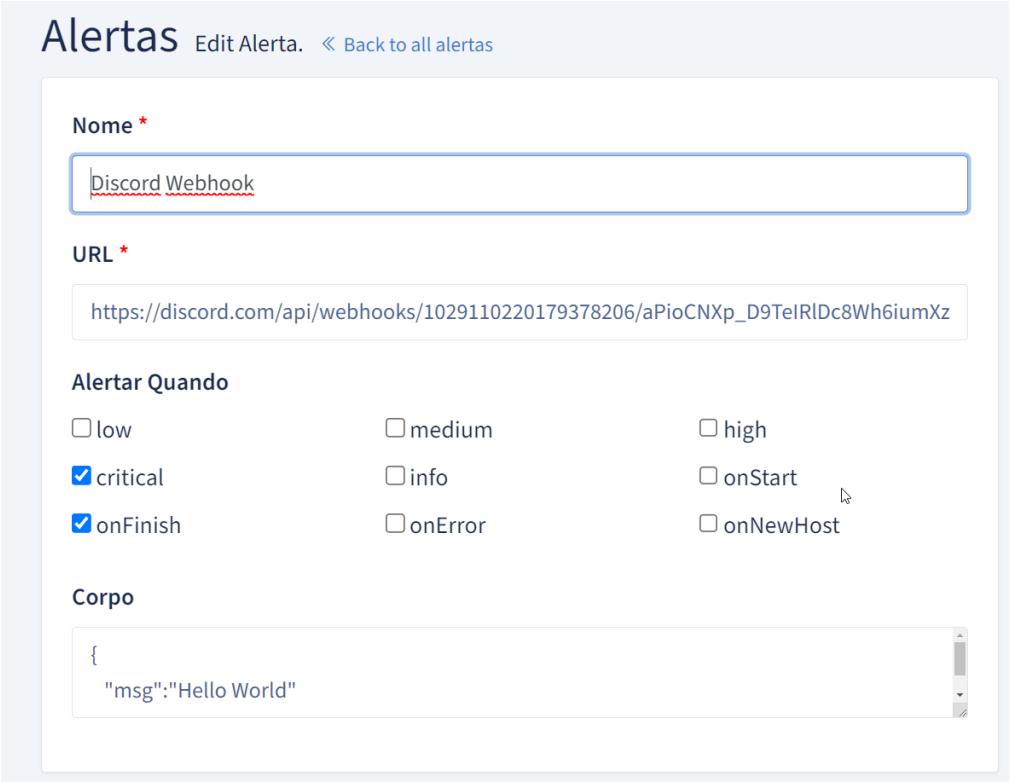
1. O usuário habilitado seleciona a opção de editar alerta na página onde os alertas são listados.

Figura 77: Selecionar a opção "Editar Alerta"



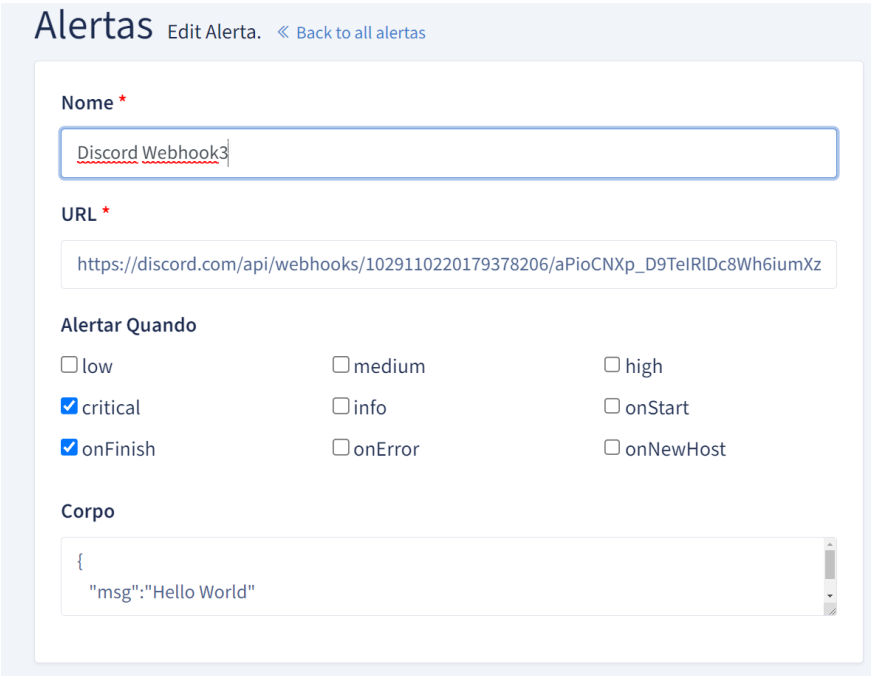
2. O sistema exibe um formulário com as informações do alerta selecionado.

Figura 78: Exibição de formulário com informações



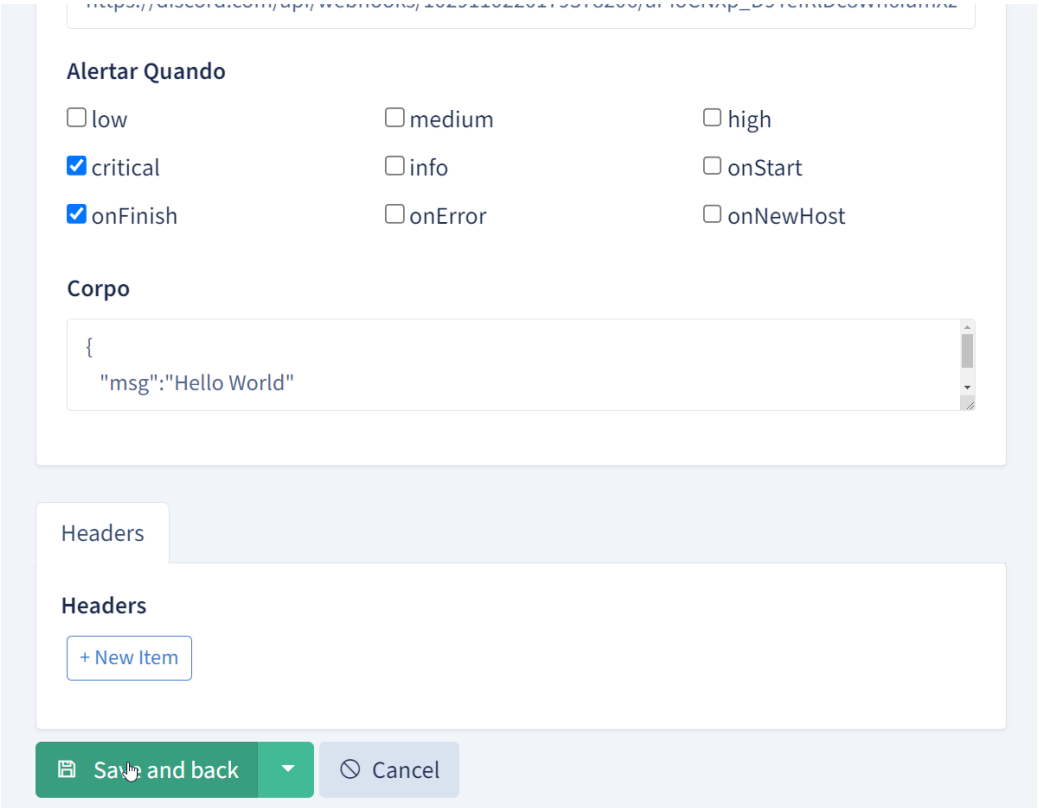
3. O usuário modifica as informações do alerta, como nome, descrição, critérios, etc.

Figura 79: Modificar as informações do alerta



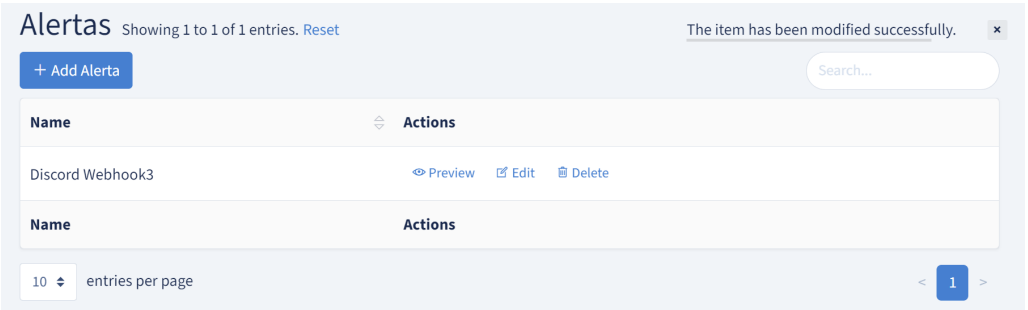
4. O usuário confirma a alteração realizada.

Figura 80: Confirmação da alteração realizada



5. O sistema atualiza as informações do alerta.
6. O usuário é redirecionado para a página de listar alertas

Figura 81: Redirecionar para a página de listar alertas



4.2.26.2 Fluxo Alternativo

1. O usuário modifica as informações do alerta com dados inválidos.
2. O sistema emite uma mensagem de erro informando que os dados são inválidos e solicita ao usuário que insira informações válidas.

3. O usuário realiza a correção dos dados e confirma a alteração novamente.

4.2.26.3 Pré-condições

1. O usuário deve estar autenticado e habilitado a editar alertas.
2. Deve haver pelo menos um alerta cadastrado no sistema Tupã.

4.2.26.4 Pós-condições

1. As informações do alerta selecionado são atualizadas no sistema.

4.2.26.4 Restrições

1. O alerta deve ter pelo menos um critério associado.
2. O nome do alerta precisa ter pelo menos 5 caracteres.

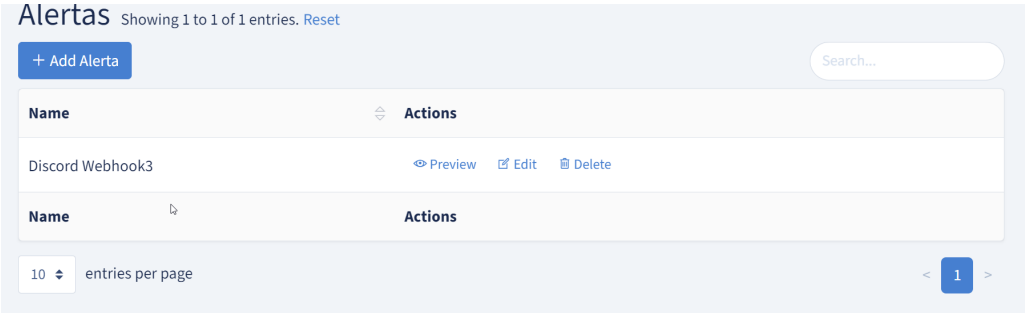
4.2.27 Excluir Alerta

Esse caso de uso permite a exclusão de um alerta previamente cadastrado no sistema Tupã.

4.2.27.1 Fluxo Básico

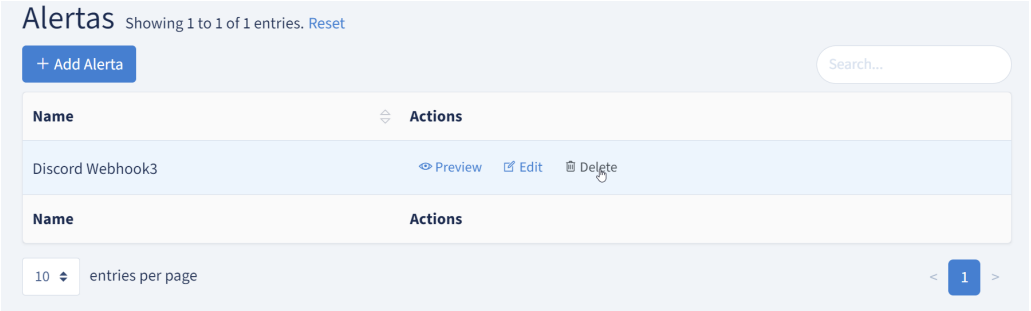
1. O usuário acessa a lista de alertas.

Figura 82: Acessar a lista de alertas



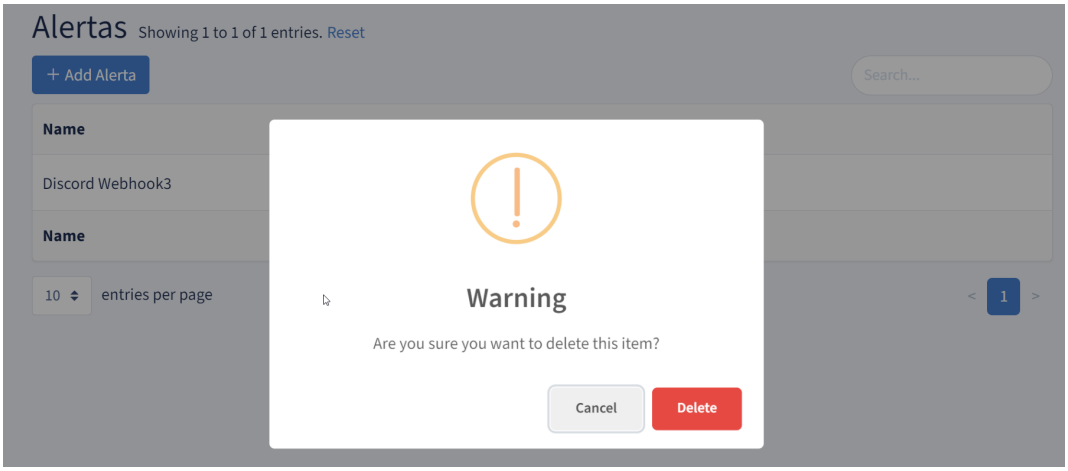
2. O usuário seleciona o botão de exclusão para o alerta desejado.

Figura 83: Selecionar a opção de exclusão



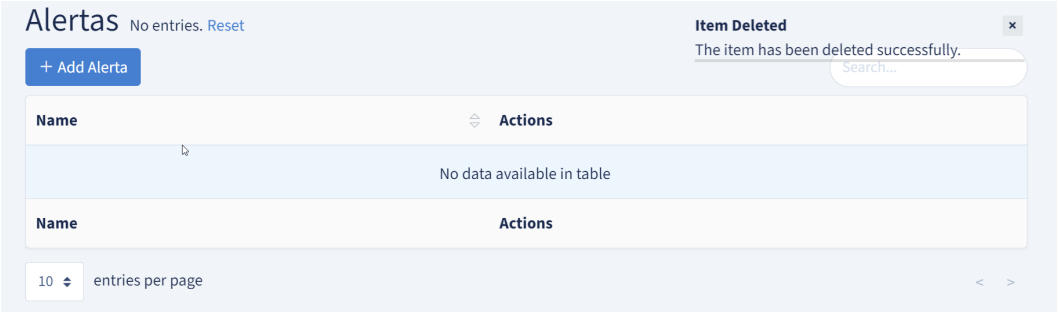
3. O sistema exibe uma mensagem de confirmação para o usuário.
4. O usuário confirma a exclusão do alerta.

Figura 84: Confirmar a exclusão



5. O sistema exibe uma mensagem de sucesso informando que o alerta foi excluído com sucesso.

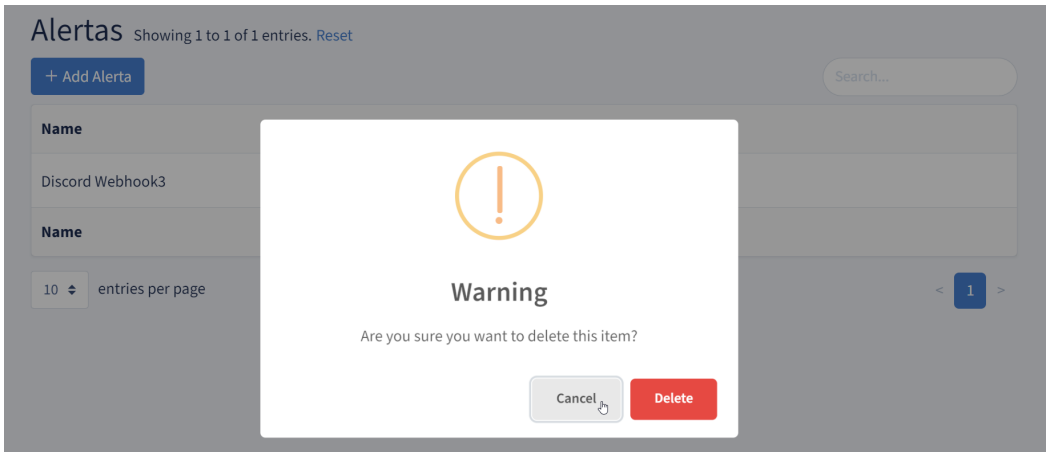
Figura 85: Alerta excluído



4.2.27.2 Fluxo Alternativo

1. Se o usuário desistir da exclusão do alerta, ele pode cancelar a operação e voltar para a página de listar alertas.

Figura 86: Ao desistir da exclusão, pode-se cancelar a operação



4.2.27.3 Pré-condições

1. Existe pelo menos um alerta cadastrado no sistema.

4.2.27.4 Pós-condições

1. O alerta selecionado é excluído do sistema.
2. O sistema volta para a página de listar alertas.

4.2.27.5 Restrições

1. Para excluir um alerta é necessário deletar todos os scans atrelados a ele.
2. O usuário precisa possuir a permissão de excluir alertas ou ser um administrador

4.2.28 Listar Alertas

Este caso de uso permite que o usuário visualize a lista de todos os alertas cadastrados no sistema Tupã.

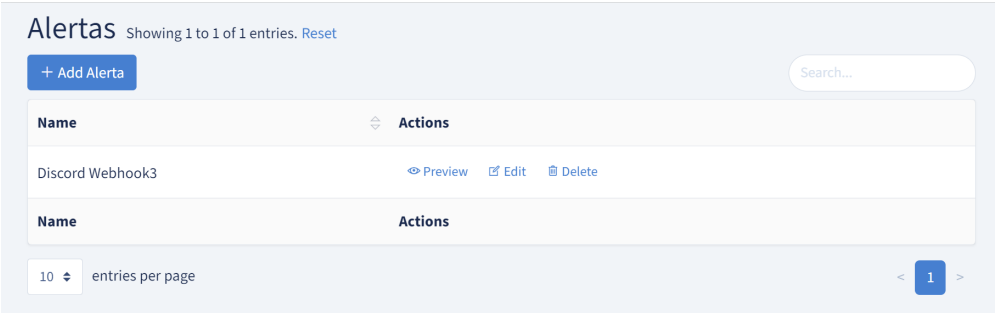
4.2.28.1 Fluxo Básico

1. O usuário habilitado seleciona a opção "Alerta" no menu.

Figura 87: Opção "Alerta" selecionada no menu

2. O sistema exibe uma tabela contendo todos os alertas cadastrados, mostrando informações como título, descrição e data de criação.

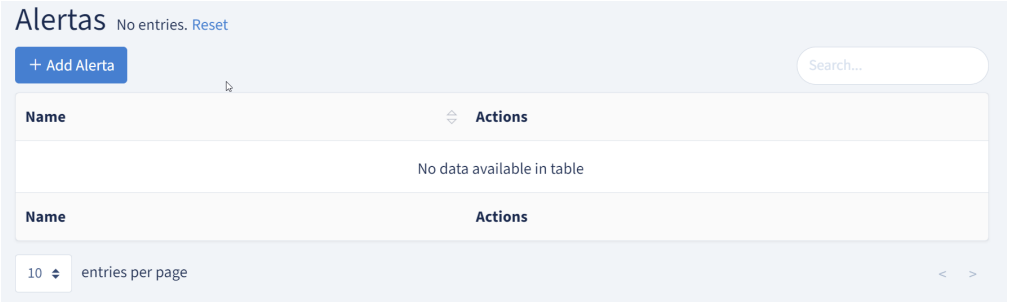
Figura 88: Sistema exibe uma tabela com as informações



4.2.28.2 Fluxo Alternativo

1. Se não houver nenhum alerta cadastrado, o sistema exibe uma mensagem informando que não existem alertas cadastrados no momento.

Figura 89: Não havendo alerta cadastrado, informa-se a ausência do mesmo



4.2.28.3 Pré-condições

1. O usuário precisa estar autenticado no Tupã.
2. Existem alertas cadastrados no sistema.

4.2.28.4 Pós-condições

1. O usuário visualiza a lista de todos os alertas cadastrados no Tupã.

4.2.28.5 Restrições

1. O usuário precisa ter a permissão de visualizar alertas ou ser administrador do sistema.

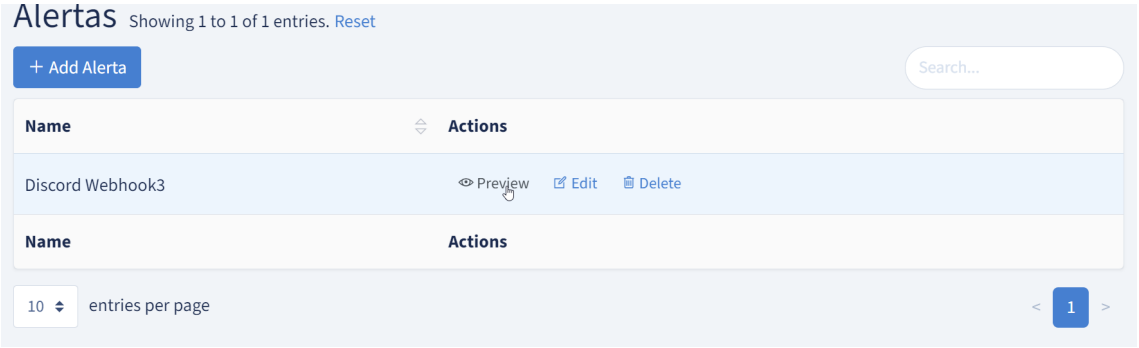
4.2.29 Mostrar Alerta

Este caso de uso permite a visualização das informações detalhadas de um alerta específico cadastrado no sistema Tupã.

4.2.29.1 Fluxo Básico

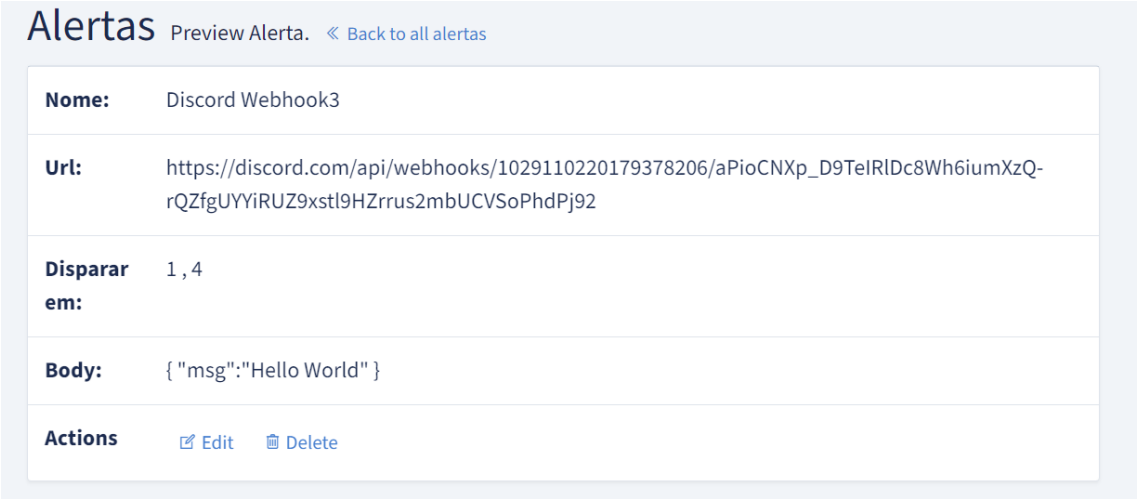
1. Em listar alertas o usuário escolhe o alerta que deseja visualizar as informações.

Figura 90: Usuário escolhe o alerta que deseja visualizar as informações



2. O sistema exibe todas as informações relacionadas ao alerta selecionado, como título, descrição e data de criação.

Figura 91: Exibição de todas as informações



4.2.29.2 Pré-condições

1. O usuário deve estar autenticado.
2. Deve haver pelo menos um alerta cadastrado no sistema Tupã.

4.2.29.3 Pós-condições

1. O usuário consegue visualizar todas as informações detalhadas do alerta selecionado.

4.2.29.4 Restrições

1. O usuário não pode editar o alerta nesta página, apenas visualizá-lo.
2. O usuário possui a permissão necessária para visualizar o alerta.

4.2.30 Fazer Login

Este caso de uso possui como objetivo permitir que o usuário faça login no sistema

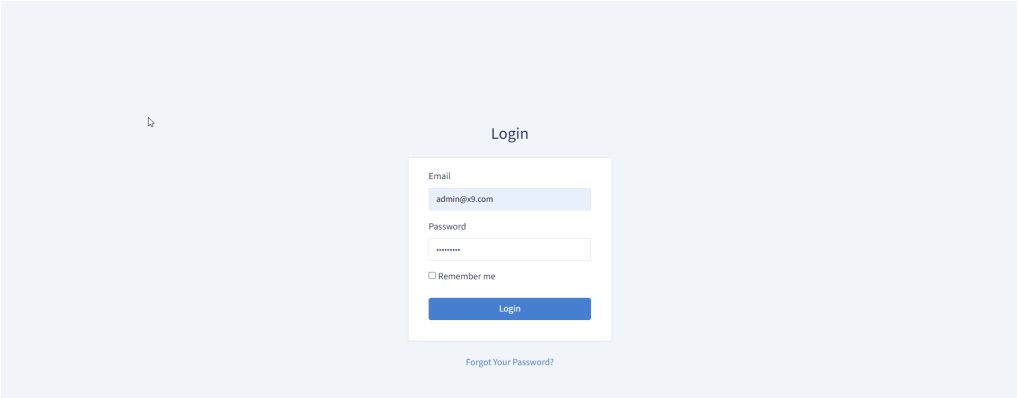
4.2.30.1 Pré-condições

1. O usuário deve estar na tela de login
2. As credenciais de login devem ser válidas

4.2.30.2 Fluxo básico

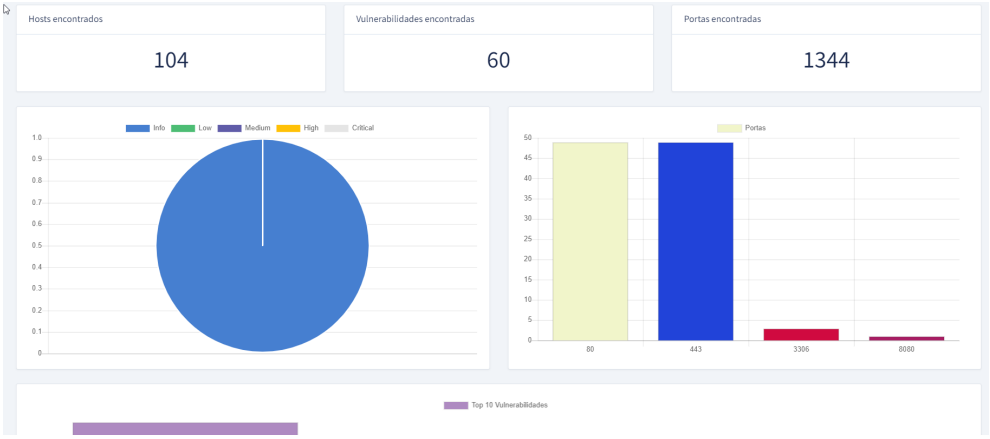
1. O usuário informa seu nome de usuário e senha

Figura 92: Informar nome e senha



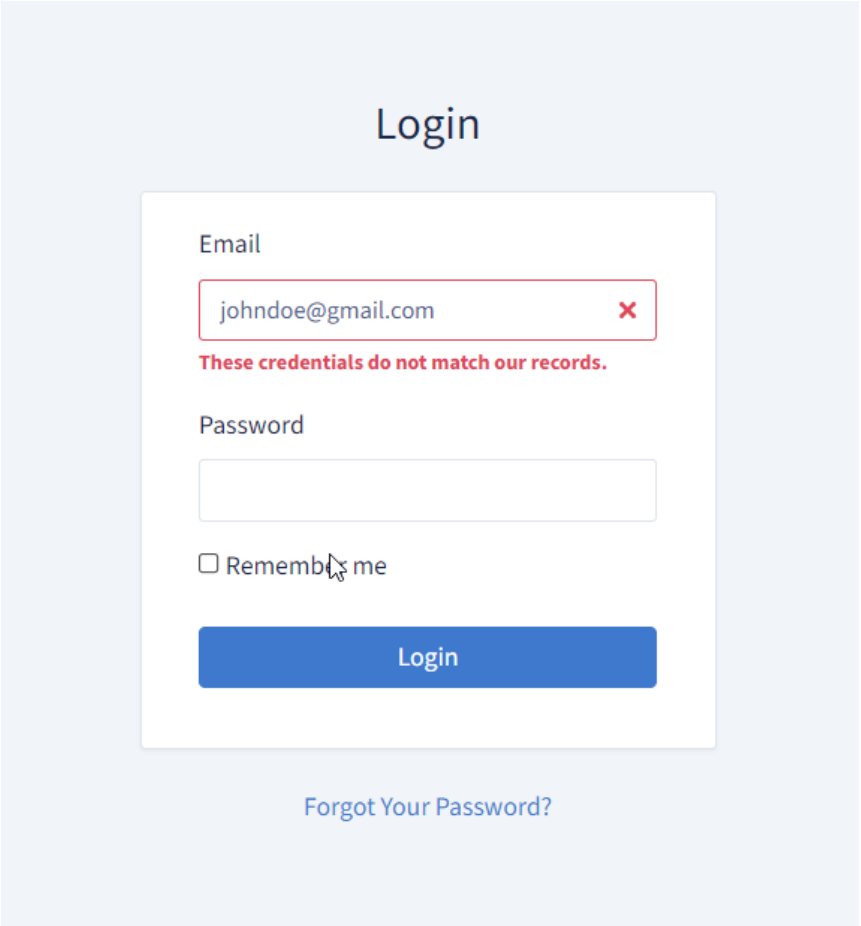
2. O sistema verifica se as credenciais são válidas
3. Se as credenciais forem válidas, o sistema redireciona o usuário para a tela inicial do sistema.

Figura 93: Com credenciais válidas, ocorre o redirecionamento do usuário para tela inicial



4. Se as credenciais forem inválidas, o sistema exibe uma mensagem de erro na tela de login.

Figura 94: Há uma exibição de erro caso as credenciais sejam inválidas



4.2.30.4 Restrições

1. As credenciais de login devem ser válidas para permitir o acesso ao sistema.

4.2.30.5 Pós-condições

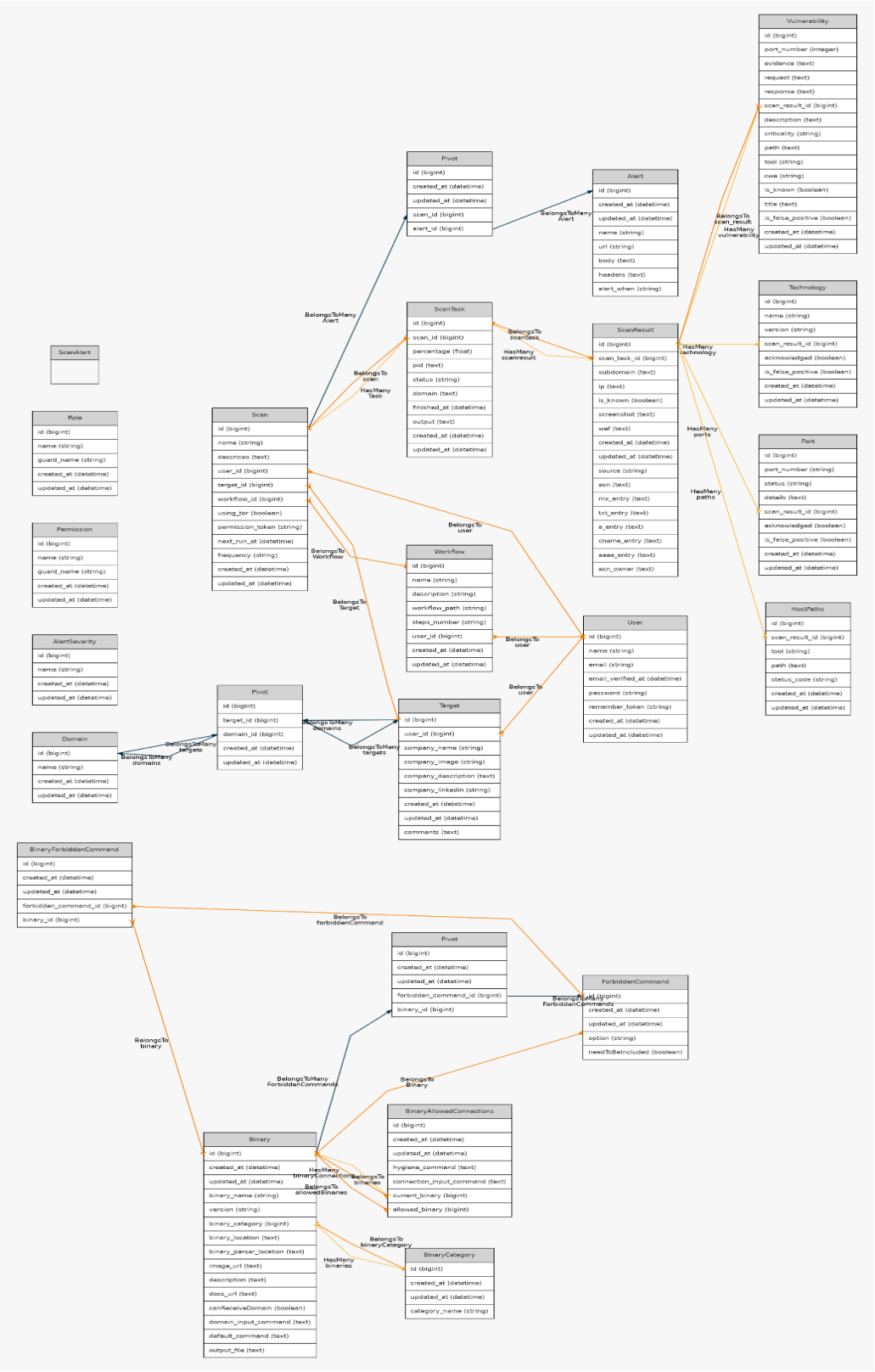
1. O usuário é autenticado no sistema e tem acesso às funcionalidades disponíveis de acordo com suas permissões.

4.3 Modelagem de dados

Para organizar os dados coletados pelos programas executados nos workflows, foi necessário construir uma base de dados utilizando o sistema de gerenciamento de banco de

dados relacional MySQL[16], na figura 95, é possível visualizar como o banco de dados está estruturado.

Figura 95: Modelo lógico do banco de dados



[16] MySQL. Disponível em: https://www.mysql.com/. Acesso em: 05 mar. 2023.

5. Projeto de Implementação do sistema Tupã

O objetivo deste capítulo é explicar de que maneira se deu a implementação do sistema, quais foram as tecnologias utilizadas, quais precauções foram tomadas para tentar garantir a confidencialidade, integridade e disponibilidade dos dados e do sistema e como o sistema funciona.

5.1 Tecnologias Utilizadas

Para o desenvolvimento da interface web e do script que vai executar as chamadas a nível de sistema operacional foi escolhida a linguagem de programação PHP [17]em sua oitava versão, com o framework Laravel [18]na versão 8. O motivo dessa escolha é que tanto a linguagem quanto o framework já são consolidados no mercado, tendo baixa probabilidade de ficarem obsoletos em um futuro próximo. Além disso, para agilizar a construção do frontend foi utilizado um outro framework chamado backpack for laravel [19]na versão 4.1 sendo sua principal função a construção de interfaces de administração.

5.2 Hardening da aplicação

No Tupã, para garantir que mesmo em um caso de eventual comprometimento do sistema, o mínimo de impacto possa ser causado, a aplicação será executada em um ambiente virtualizado com o mínimo de permissão e programas necessários para o bom funcionamento do software. Para a virtualização do ambiente foi escolhida a tecnologia Docker[20].

[17] PHP. Disponível em: https://www.php.net/. Acesso em: 5 mar. 2023
[18] LARAVEL. Laravel - The PHP Framework For Web Artisans. [S.l.], [s.d.]. Disponível em: https://laravel.com/. Acesso em: 07 mar. 2023.
[19] BACKPACK FOR LARAVEL. Disponível em: https://backpackforlaravel.com/. Acesso em: 07/03/2023.
[20] DOCKER. In: Docker [S.l.]. Disponível em: https://www.docker.com/. Acesso em: 24 fev. 2023.

5.3 Workflows

O objetivo deste tópico é explicar com mais detalhes qual é o papel dos workflows neste trabalho e de quais maneiras é possível criar um.

5.3.1 Engine

A engine escolhida para ser responsável pela execução dos workflows é o Digdag[21]. O Digdag é uma ferramenta de código aberto que tem como funcionalidade, ajudar a criar, executar e monitorar tarefas automatizadas complexas. Para tal utiliza Grafos Acíclicos Diretos como infraestrutura. É muito utilizado na área de desenvolvimento de software para coleta e transformação de dados em aplicações que possuem dados em diferentes ferramentas de coleta de logs.

5.3.2 SDK Heimdall

Um SDK[22], também chamado de software development kit (kit de desenvolvimento de software em português) é qualquer recurso disponibilizado por um fabricante ou empresa com o intuito de acelerar a criação ou integração de um produto com outros produtos criados por desenvolvedores.

O objetivo de criar um SDK para integrar workflows através de código é reduzir o tempo necessário para usar novas ferramentas. Além disso, criar uma camada intermediária entre o programador e as APIs, para facilitar o envio de informações para o Tupã. O Python foi escolhido como a linguagem para o SDK, devido à sua versatilidade e facilidade de uso, sendo muito popular entre aqueles que possuem pouca experiência em programação de software.

O SDK é baseado em classes, essas por sua vez possuem uma lógica em sua nomenclatura. As classes base do SDK, que servem para realizar a integração com o Tupã, possuem a lógica Base{Objeto de Envio}Parser. Essa sintaxe ajuda a indicar qual classe deve ser estendida para a implementação de uma nova ferramenta. Por exemplo, para que o Tupã consiga ler a saída da ferramenta Subfinder que é uma ferramenta de enumeração de subdomínios. Faremos uma nova classe de nome “Subfinder”, sendo esta filha da classe BaseSubdomainParser, como pode ser vista no exemplo demonstrado na figura 96:

Figura 96: Implementação do subfinder no Tupã

[21] Link para o digdag: https://www.digdag.io/
[22] Auth0. What Is an SDK? Disponível em: https://auth0.com/blog/what-is-an-sdk/. Acesso em: 07 mar. 202

```python
from modules.Heimdall import Heimdall
from classes.BaseSubdomainParser import BaseSubdomainParser
import sys, json

class Subfinder(BaseSubdomainParser):

    def __init__(self, arguments) -> None:
        super().__init__(arguments)

        for line in sys.stdin:
            heimdall = Heimdall(self.getApiPath(), self.getToken(), self.getScanId(), self.getStepNumber())
            payload = json.loads(line.strip())
            self.setSubdomain(payload['host'])
            self.setSource(payload['source'])

            heimdall.sendSubdomain(self.getSubdomain(), self.getSource())
```

Na figura 96 é possível notar que excluindo as linhas em que as bibliotecas são importadas para o arquivo, são necessárias pouco mais de 10 linhas para a implementação do Subfinder ao Tupã. O que faz com que a implementação de uma nova ferramenta leve poucos minutos.

Depois da manipulação da saída da ferramenta, é necessário instanciar um objeto da classe Heimdall e chamar a função correspondente para o envio da informação, todas elas seguem a sintaxe send{Objeto a ser enviado} e possuem em suas assinaturas descrição e quais argumentos ela espera. Uma vez com a classe da ferramenta implementada, basta apenas que o profissional crie um arquivo chamado allowedTools.yaml e insira o nome da classe criada em uma das categorias disponíveis no documento.

Figura 97: Arquivo allowedTools.yaml

```yaml
AllowedSubdomainTools:
  - subfinder
AllowedVulnerabilityTools:
  - nuclei
  - zap
  - dalfox
AllowedTechnologiesIdentifierTools:
  - nucleitech
AllowedWafTools:
  - wafw00f
AllowedPortScanningTools:
  - Nmap
AllowedCrawlingTools:
  - katana
```

Além da classe BaseSubdomainParser que possui como utilidade facilitar a integração de ferramentas de enumeração de subdomínios o Heimdall conta com outras classes para a integração de ferramentas que possuem propósitos diferentes:

- BaseVulnerabilityParser: Classe indicada para a integração de scanners de vulnerabilidades.
- BasePathParser: Classe indicada para a integração de crawlers de websites.
- BasePortParser: Classe indicada para a integração de scanners de portas.
- BaseTechnologyParser: Classe indicada para a integração de identificadores de tecnologia;
- BaseWafParser: Classe indicada para a integração de Identificadores de Web Application Firewalls.

Além das classes mencionadas anteriormente, o Tupã também conta com outras classes auxiliares para manipulação de dados que são comuns em ferramentas de segurança:

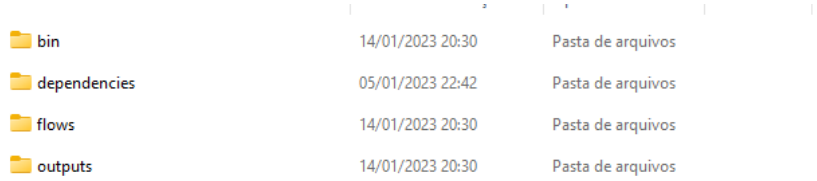
- IP: Indicada para manipular e obter endereços de rede de aplicações web.
- Port: Indicada para manipular portas encontradas
- Subdomain: Indicada para manipular subdomínios encontrados
- Technology: Indicada para manipular Tecnologias encontradas
- Waf: Indicada para manipular Wafs encontrados.

É possível integrar uma ferramenta sem utilizar nenhuma das classes disponibilizadas no SDK, desde que o nome da classe seja o mesmo da ferramenta e esteja no arquivo YAML allowedTools. Basta apenas utilizar as rotas que serão mencionadas na seção 5.5. Apesar disso, o uso do SDK é fortemente indicado para fins de padronização e aumento de produtividade.

5.4.3 Criando um workflow

Existem duas maneiras para criar um workflow no Tupã. A primeira delas é através de código, baixando o template para a criação de workflow disponível no próprio Tupã. Para realizar configuração de um workflow programaticamente, foi definido por esse trabalho a seguinte estrutura de arquivos:

Figura 98: Estrutura de pasta criar Workflow



Na pasta flows estão localizados os arquivos .dig necessários para a execução do workflow. Para que ele funcione adequadamente ao realizar um scan é sempre necessário que exista um arquivo de nome start.dig.

Os arquivos .dig seguem uma sintaxe específica que pode ser vista com mais detalhes na documentação do Digdag. O exemplo de workflow programático disponibilizado no Tupã contém alguns arquivos .dig comentados explicando o que cada linha significa.

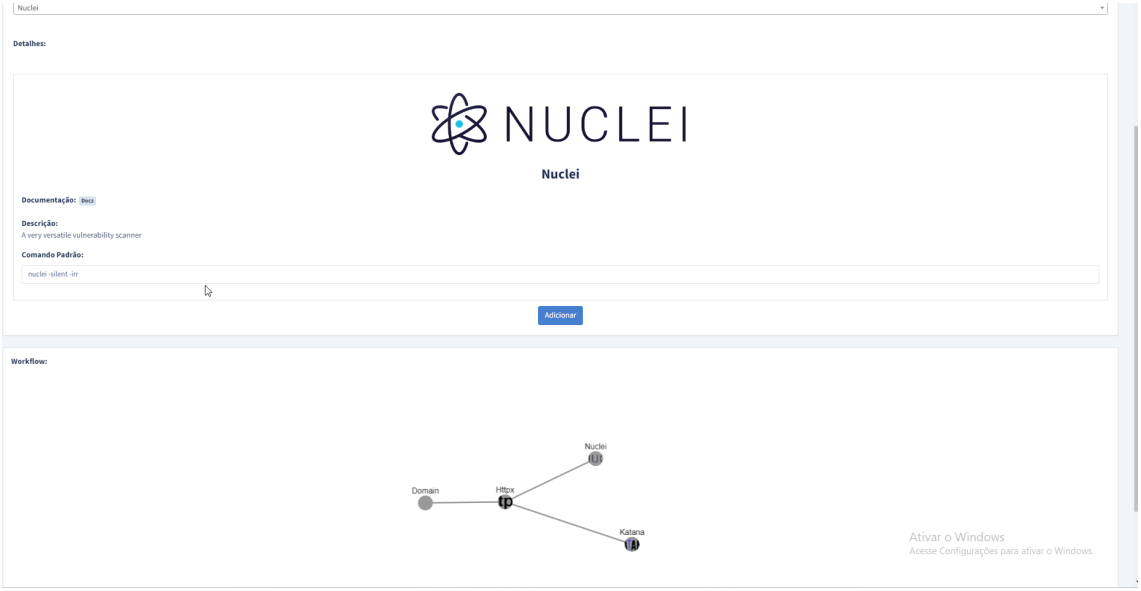
Na pasta bin, temos todo e qualquer programa utilizado pelo workflow, além do SDK mencionado no subitem anterior.

Para habilitar a transferência dos dados de uma ferramenta para o Tupã, basta copiar as classes com o nome da ferramenta junto com o arquivo allowedTools.yaml para a pasta parsers.

Na pasta dependencies ficam todo e qualquer arquivo necessário para utilização dos binários (bibliotecas auxiliares, dlls, wordlists entre outros…) Por último, a pasta outputs serve para guardar as saídas das ferramentas utilizadas.

A segunda maneira é utilizar a interface gráfica para criação de workflows presente no Tupã. Esse, por sua vez, carrega uma série de parsers e ferramentas pré-implementadas para aqueles que não possuem tempo ou conhecimento suficiente das ferramentas para criar workflows programaticamente.

Figura 99: Interface gráfica para criação de Workflow



Basta apenas selecionar a ferramenta, personalizar os comandos, caso seja um usuário intermediário, e conectar com a ferramenta anterior desejada, formando assim, um grafo.

5.5 Rotas de api

Além da interface web, o Tupã conta também com um sistema de Apis. É através delas que a interface web é capaz de receber os resultados das ferramentas. Logo, podem ser utilizadas pelos usuários do sistema tanto para caso estes desejem criar workflows programaticamente sem a ajuda de uma abstração(SDK Heimdall), quanto para caso estes desejem fazer mudanças no mesmo SDK. As rotas de Api são:

Tabela 3: Rota setProgress

Rota	/api/setProgress
Método	POST
Descrição	Endpoint para inserir o valor do progresso de um scan

Tabela 4: Rota domains

Rota	/api/domains
Método	POST

Descrição	Endpoint para inserir novos domínios em um determinado scan

Argumentos	"scan_id": {
	"tipo": "*string*",
	"nullable": *false*,
	"formato": "*string*"
	},
	"domains": {
	"tipo": "*array*",
	"nullable": *false*,
	"formato": "*array*"
	},
	"steps_number": {
	"tipo": "*integer*",
	"nullable": *false*,
	"formato": "*integer*"
	}

Tabela 5: Rota Vulnerability

Rota	/api/vulnerability
Método	POST
Descrição	Endpoint para inserir novas vulnerabilidades em um host identificado pertencente a um determinado scan

Argumentos	"scan_id": {
	"tipo": "*integer*",
	"nullable": *false*,
	"formato": "*integer*"
	},
	"port": {

```
    "tipo": "integer",
    "nullable": true,
     "formato": "integer"
},
"severity": {
       "tipo": "string",
       "nullable": false,
       "formato": "string"
},
"path": {
        "tipo": "string",
         "nullable": false,
         "formato": "string"
},
"tool": {
          "tipo": "string",
          "nullable": false,
          "formato": "string"
},
"vulnerable_ip": {
          "tipo": "string",
          "nullable": true,
          "formato": "string"
},
"vulnerable_domain": {
           "tipo": "string",
          "nullable": true,
          "formato": "string"
},
"description": {
           "tipo": "string",
           "nullable": true,
           "formato": "string"
```

```
 },
"cwe": {
            "tipo": "string",
           "nullable": true,
           "formato": "string"
},
"steps_number": {
            "tipo": "integer",
            "nullable": false,
             "formato": "integer"
},
"title": {
            "tipo": "string",
            "nullable": false,
             "formato": "string"
},
"evidence": {
             "tipo": "string",
             "nullable": true,
             "formato": "string"
 },
"request": {
                    "tipo": "string",
                   "nullable": true,
                  "formato": "string"
 },
"response": {
                     "tipo": "string",
                    "nullable": true,
                   "formato": "string"
 }
```

Tabela 6: Rota Ports

Rota	/api/ports
Método	POST
Descrição	Endpoint para inserir novas portas abertas em um domínio previamente encontrado

Argumentos	"scan_id": {
	"tipo": "string",
	"nullable": false,
	"formato": "string"
	},
	"ports": {
	"tipo": "string",
	"nullable": false,
	"formato": "string"
	},
	"domain": {
	"tipo": "string",
	"nullable": true,
	"formato": "string"
	},
	"ip": {
	"tipo": "string",
	"nullable": true,
	"formato": "string"
	},
	"steps_number": {
	"tipo": "integer",
	"nullable": false,
	"formato": "integer"
	}

Tabela 7: Rota Waf

Rota	/api/waf
Método	POST
Descrição	Endpoint para inserir no domínio qual tipo de WAF (Web Application Firewall) ele utiliza em caso de utilização de alguma ferramenta para esse fim

Argumentos	"waf": {
	"tipo": "string",
	"nullable": false,
	"formato": "string"
	},
	"scan_id": {
	"tipo": "string",
	"nullable": false,
	"formato": "string"
	},
	"subdomain": {
	"tipo": "string",
	"nullable": false,
	"formato": "string"
	},
	"steps_number": {
	"tipo": "integer",
	"nullable": false,
	"formato": "integer"
	}

Tabela 8: Rota Technology

Rota	/api/technology

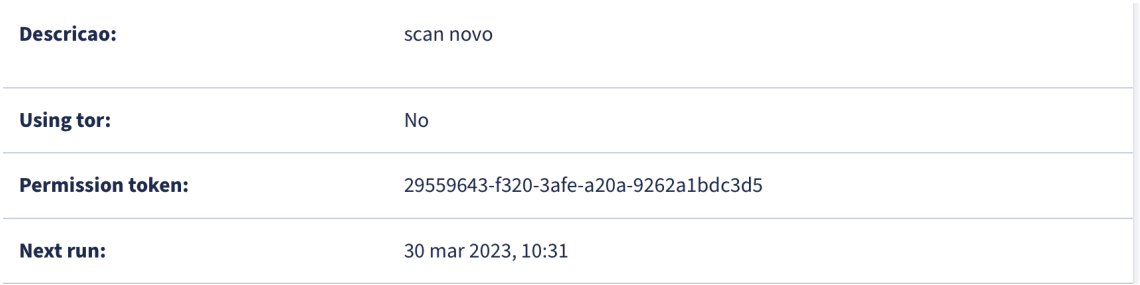
Método	POST
Descrição	Endpoint para inserir novas tecnologias identificadas em um host

Argumentos	"name": {
	"tipo": "string",
	"nullable": false,
	"format": "string"
	},
	"version": {
	"tipo": "string",
	"nullable": true,
	"format": "string"
	},
	"scan_id": {
	"tipo": "string",
	"nullable": false,
	"format": "string"
	},
	"subdomain": {
	"tipo": "string",
	"nullable": false,
	"format": "string"
	},
	"ip": {
	"tipo": "string",
	"nullable": true,
	"format": "string"
	}

5.5.1 Enviando requisições para as Apis do Tupã

Para enviar requisições para a api do Tupã é necessário passar um header de nome Authorization junto com a requisição a ser feita. O valor do header deve ser o UUID presente no campo "Permission Token" no Scan para qual o usuário deseja enviar informações.

Figura 100: Exemplo de permission token



6.Experimentação

O objetivo deste capítulo é realizar uma experimentação de forma a verificar quais resultados ela é capaz de obter em um cenário próximo do real.

6.1 Alvo

Para verificar a capacidade da ferramenta em um ambiente real, resolvemos realizar o teste no domínio raiz **uniriotec.br** e em todos os seus subdomínios, com o objetivo de encontrar e reportar falhas de segurança no ambiente do CCET/Unirio.

Para tal, as seguintes ferramentas foram integradas ao Tupã:

- Nuclei: Scanner de vulnerabilidades open source
- Httpx[23]: Uma "caixa de ferramentas" para o protocolo Http
- Katana[24]: Um crawler para websites
- Subfinder: Ferramenta que encontra subdomínios baseado em um domínio raiz
- Nmap: Scanner de portas de um host

6.2 Resultado do Scan

O scan foi realizado no dia 23 de janeiro de 2023 às 11h e terminou às 13h, e foram encontrados 104 subdomínios. A lista completa dos subdomínios e das vulnerabilidades encontradas pode ser vista no apêndice B. Sobre as vulnerabilidades encontradas foram encontradas ao todo 607 vulnerabilidades sendo elas:

- 9 de categoria média
- 21 de categoria baixa
- 577 de categoria informacional

[23] PROJECTDISCOVERY. Httpx. Disponível em: https://github.com/projectdiscovery/httpx. Acesso em: 03 mar. 2023.
[24] PROJECTDISCOVERY. Katana. Disponível em: https://github.com/projectdiscovery/katana. Acesso em: 03 mar. 2023.

Nem todas as vulnerabilidades de categoria informacional representam, de fato, alguma ameaça e, por isso, ressaltamos apenas as mais relevantes neste capítulo. Para mais informações verifique o apêndice B.

Todas as vulnerabilidades de categoria média tratam-se de vulnerabilidades de cross-site-scripting refletido, causado pela utilização do plugin para wordpress chamado Elementor. O Cross-Site-Scripting, também chamado de XSS, é um tipo de vulnerabilidade de segurança em aplicações web que permite que invasores sejam capazes de injetar código malicioso (em sua maioria javascript) em páginas web com a intenção de alterar seu comportamento padrão. Essa vulnerabilidade pode ser utilizada para roubo de dados do usuário ou para alterar o conteúdo da página.

As vulnerabilidades de categoria baixa estão divididas em:

1. Certificado SSL desencaixado
2. Certificado SSL expirado
3. Exposição do arquivo php.info
4. Exposição do arquivo behat.yaml
5. Exposição do arquivo php.ini

O certificado SSL desencaixado é quando o certificado SSL que está sendo utilizado para um determinado endereço, não corresponde ao domínio para o qual ele foi emitido, ou então, o certificado não foi emitido por uma autoridade certificadora confiável, o que torna um atacante capaz de interceptar e decodificar os dados passados do site para a rede.

O certificado SSL expirado é uma vulnerabilidade em que o certificado daquele site em específico passou de sua data de validade, o que torna possível um atacante interceptar e decodificar as comunicações entre a máquina que acessou o site e o servidor que hospeda o sistema.

Por sua vez, a exposição do arquivo php.info ocorre quando, por falta de configuração, o arquivo fica disponível para acesso público na internet. Essa exposição pode permitir que atacantes obtenham informações valiosas sobre a aplicação, como a versão do PHP utilizada, extensões em uso e, em alguns casos, até mesmo senhas de banco de dados.

Da mesma maneira, a exposição do arquivo behat.yml se dá quando esse arquivo é disponibilizado publicamente na internet, permitindo que informações sensíveis sobre a configuração de testes automatizados, como credenciais de acesso a bancos de dados, URLs e senhas, sejam acessadas por pessoas mal-intencionadas

Das vulnerabilidades de categoria info destacam-se as seguintes:

1. Arquivo de mudanças do Moodle
2. Wordpress com Xmlrpc.php habilitado
3. Exposição do arquivo Node shrinkwrap
4. Exposição do arquivo composer.json
5. Diretório .idea exposto
6. Web.config disponível para download
7. Falta de headers de segurança

Quando o arquivo de mudanças do Moodle é exposto publicamente, ele pode revelar informações sobre as configurações e personalizações do sistema Moodle, tornando possível para os atacantes identificar e explorar vulnerabilidades conhecidas.

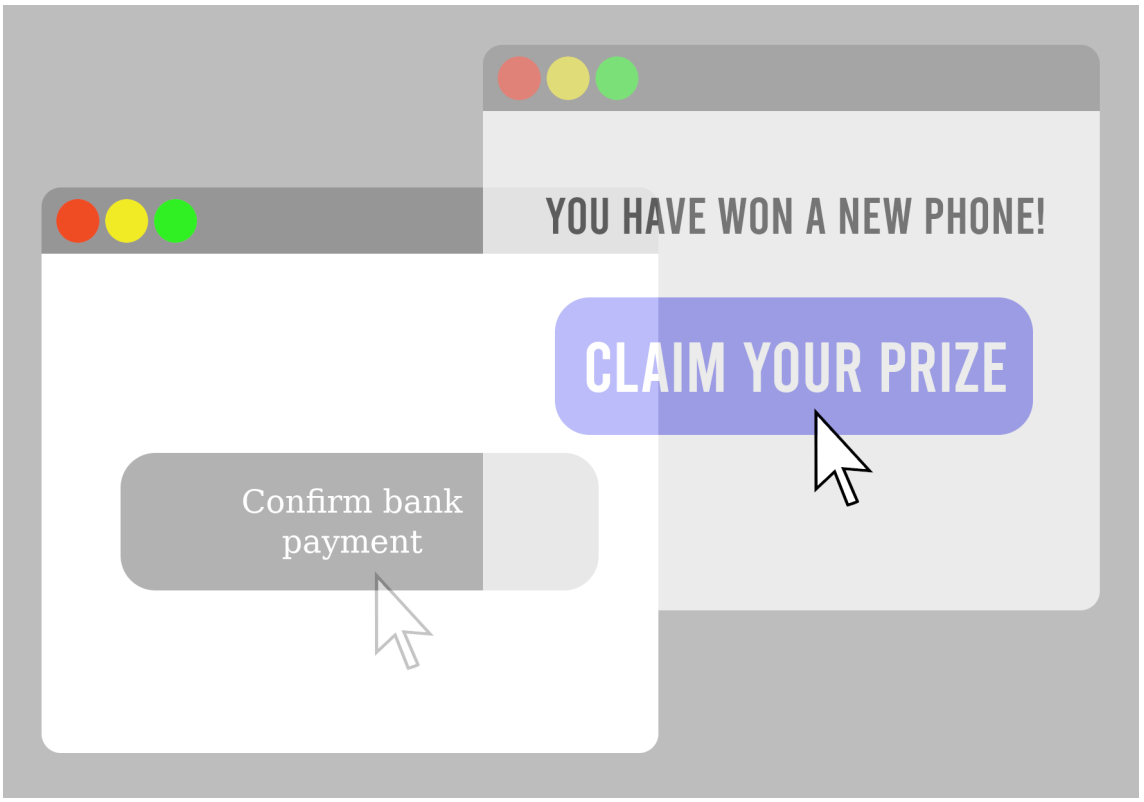
A habilitação do arquivo xmlrpc.php em um site WordPress, pode ser utilizada para a exploração por atacantes para obter informações sobre o sistema, como usuários e senhas, para realizar ataques de força bruta, escanear portas da rede interna e até ataques de negação de serviço usando o site vulnerável como nó de uma botnet através do método Pingback. Este, por sua vez, é um método presente no xmlrpc que permite que um usuário faça requisições para um outro site utilizando a aplicação wordpress como proxy.

Quando os arquivos node shrinkwrap, composer.json, web.config e o diretório .idea são expostos publicamente, possuem informações sobre as dependências do sistema e as versões utilizadas nele, o que pode ser usado por atacantes para identificar vulnerabilidades conhecidas.

Com a utilização de headers corretos de segurança é possível obter alguma blindagem contra alguns tipos de ataque, sendo alguns deles o já mencionado Cross-Site-Scripting e outros como CSRF (Cross-Site-Request-Forgery) que é quando um atacante se aproveita de uma sessão já ativa de um usuário para, através de alguma engenharia social e de programação, executar ações maliciosas em um site. Como por exemplo, trocar a senha de um usuário ou realizar uma transferência bancária.

Outra vulnerabilidade que pode ser impedida com o uso de headers de segurança é o Clickjacking, que ocorre através de engenharia social, quando um atacante convence um usuário a realizar uma ação não intencional utilizando uma Interface de usuário enganosa por cima da interface legítima do site. A figura 152 é um exemplo de como o ataque de “Clickjacking” funciona.

Figura 101: Demonstração de ataque de “clickjacking”



6.3 Análise da Ferramenta

O objetivo desta seção é mostrar que o Tupã foi capaz de cumprir cada um dos objetivos elicitados no capítulo 3.

Os objetivos descritos no capítulo 3 eram que a ferramenta estivesse de acordo com os critérios baseados no MITRE mencionados e que, ao mesmo tempo, fosse capaz de sanar o problema de ter que analisar a saída de múltiplas vulnerabilidades.

Diante dos resultados expostos na seção **6.2 Resultado do Scan,** dos casos de uso apresentados no capítulo **4 Especificações do Sistema** e das informações dispostas no relatório disponibilizado no apêndice **B** podemos chegar a conclusão de que o Tupã, conseguiu concluir cada um dos objetivos propostos.

No caso de uso do “ Vizualizar Resultado do Scan”, é possível visualizar que ele é capaz de receber, portas, subdomínios e vulnerabilidades.

No apêndice **B** é possível visualizar um relatório gerado através do Tupã de maneira automatizada.

Na seção **6.2** é possível perceber que é possível integrar diferentes ferramentas e obter um resultado único, comprovando que o Tupã consegue resolver o problema das múltiplas

entradas de arquivo, desde que haja a interação entre a ferramenta e as APIs do Tupã, seja por meio do SDK Heimdall ou por interação direta.

7. Conclusão

O objetivo deste capítulo é propor uma conclusão a respeito do resultado deste trabalho, suas limitações e por último apresentar as possibilidades de expansão através de trabalhos futuros.

7.1 Considerações

O objetivo deste trabalho era construir uma aplicação que fosse capaz de executar automaticamente alguns dos passos contidos no MITRE PRE ATT&CK e gerar relatórios de maneira rápida e dinâmica. Diante dos resultados obtidos, é possível notar que esse objetivo foi alcançado com sucesso. Embora ainda modesto frente aos seus pares disponíveis no mercado, o Tupã pode representar uma alternativa nacional e mais barata se comparada a outras aplicações do mesmo tipo desenvolvidas por empresas de outros países. Além disso, devido à sua modularidade é facilmente possível alcançar resultados melhores se novas ferramentas de análise de segurança forem adicionadas aos workflows já existentes.

7.2 Limitações no trabalho

Sobre as limitações no trabalho, a maior delas foi a impossibilidade de explorar algumas das vulnerabilidades encontradas, visto que elas poderiam causar indisponibilidade no sistema e como o CCET, que foi o alvo de teste em questão não possui ambientes de espelho, optamos por não seguir adiante com nenhuma exploração

7.3 Limitações da Ferramenta

Apesar do resultado positivo, o Tupã ainda apresenta algumas limitações, como a engine de execução de workflows não ser proprietária, além disso há a necessidade de algum conhecimento prévio sobre o uso de cada uma das ferramentas que serão utilizadas pelo sistema.

Além disso, é preciso lembrar que apenas a utilização do Tupã não servirá para manter uma empresa segura, o objetivo deste trabalho não é desenvolver uma solução final e definitiva para segurança da informação. Logo, seu uso deve vir em conjunto com outras

práticas de segurança, como a aplicação de *patches* de correção de vulnerabilidades e a implementação de políticas de desenvolvimento seguro.

7.4 Trabalhos futuros

O caminho mais óbvio seria criar a própria engine para execução dos workflows, de maneira a ser mais rápida que o digdag. Além disso, desenvolver um programa de linha de comando que facilite ainda mais a criação de um workflow de maneira programática. Outra possibilidade seria levar o scanner até a iniciativa privada e verificar como seria sua performance em ambientes corporativos.

Referências Bibliográficas

RYDNING, David Reinsel-John Gantz-John; REINSEL, J.; GANTZ, J. The digitization of the world from edge to core. **Framingham: International Data Corporation**, v. 16, 2018.

CHNG, Samuel et al. Hacker types, motivations and strategies: A comprehensive framework. **Computers in Human Behavior Reports**, v. 5, p. 100167, 2022.

PRADO, Filipe. Brasil foi 5º país com mais ataques cibernéticos no ano. Istoedinheiro, [*S. l.*], p. 1-1, 21 dez. 2021. Disponível em: https://www.istoedinheiro.com.br/brasil-foi-5o-pais-com-mais-ataques-ciberneticos-no-ano-relembre-os-principais/. Acesso em: 9 jun. 2022

BUCKLAND, Michael Keeble. **Information and information systems**. ABC-CLIO, 1991.

MICHAELIS UOL. Informação. Michaelis UOL, n.d. Disponível em: https://michaelis.uol.com.br/moderno-portugues/busca/portugues-brasileiro/informa%C3%A7%C3%A3o/. Acesso em: 28 fev. 2023

HUMBY, C. Address to the ANA senior marketers' summit, kellogg school of management, northwestern university. **Data is the new oil. Retrieved from http://ana. blogs. com/maestros**, 2006.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **ABNT NBR ISO/IEC 17799: 2005: tecnologia da informação-técnicas de segurança-código de prática para a agestão da segurança da informação**. ABNT, 2005.

AGENCE FRANCE-PRESSE. Renault revela caso interno de espionagem grave que ameaçava ativos estratégicos. UOL Carros, São Paulo, 06 jan. 2011. Disponível em: https://www.uol.com.br/carros/noticias/afp/2011/01/06/renault-revela-caso-interno-de-espionagem-grave-e-que-ameacava-ativos-estrategicos.htm. Acesso em: 24 fev. 2023.

SECTOR, STANDARDIZATION; ITU, O. F. Itu-tx. 1205. **Interfaces**, v. 10, n. 20-X, p. 49.

BUSH, George. National Security Presidential Directive 54: Cybersecurity Policy. 2008.

EQUIFAX to pay up to $700m to settle data breach. BBC, [*S. l.*], p. 1, 22 jul. 2019. Disponível em: https://www.bbc.com/news/technology-49070596. Acesso em: 24 fev. 2023.

SANTOS, Joanna CS; TARRIT, Katy; MIRAKHORLI, Mehdi. A catalog of security architecture weaknesses. In: **2017 IEEE International Conference on Software Architecture Workshops (ICSAW)**. IEEE, 2017, p. 1

BARKER, Elaine et al. A profile for us federal cryptographic key management systems. **NIST Special Publication**, v. 800, p. 152, 2014.

USC, Sec. 3502. **CNSSI-4009**.p.173

CHAPLAIN, Cristina. Weapon Systems Cybersecurity: DoD just beginning to grapple with scale of vulnerabilities. **Washington, DC, USA, GAO Report No. GAO-19-128**, 2018.

CVE. FAQs. Disponível em: https://www.cve.org/ResourcesSupport/FAQs#pc_introcve_nvd_relationship. Acesso em: 27 fev. 2023.

MITRE Corporation. CWE - Common Weakness Enumeration. Disponível em: https://cwe.mitre.org. Acesso em: 27 fev. 2023.

MITRE Corporation. ATT&CK Matrix for Enterprise - Pre-Attack. Disponível em: https://attack.mitre.org/matrices/enterprise/pre/. Acesso em: 27 fev. 2023

LOCKHEED MARTIN. Cyber Kill Chain®. Disponível em: https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html. Acesso em: 05 mar. 2023.

Apêndice A - Descrições de casos de uso

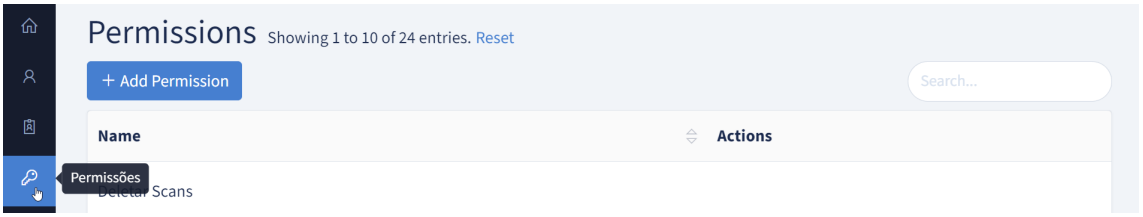
Criar Permissão

Este caso de uso permite a criação de uma nova permissão para um papel no sistema Tupã.

Fluxo Básico

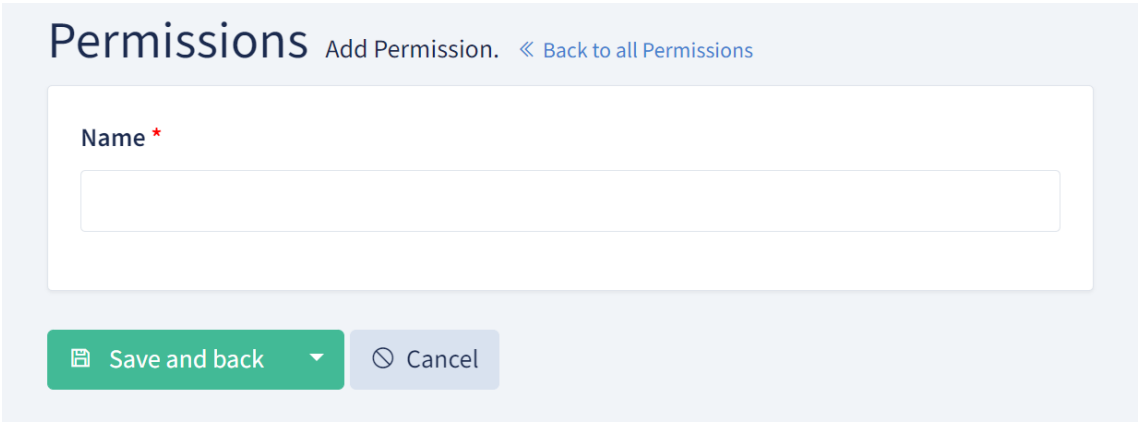
1. O administrador seleciona a opção de listar permissão.

Figura 102: Selecionar a opção "Listar Permissão" pelo administrador



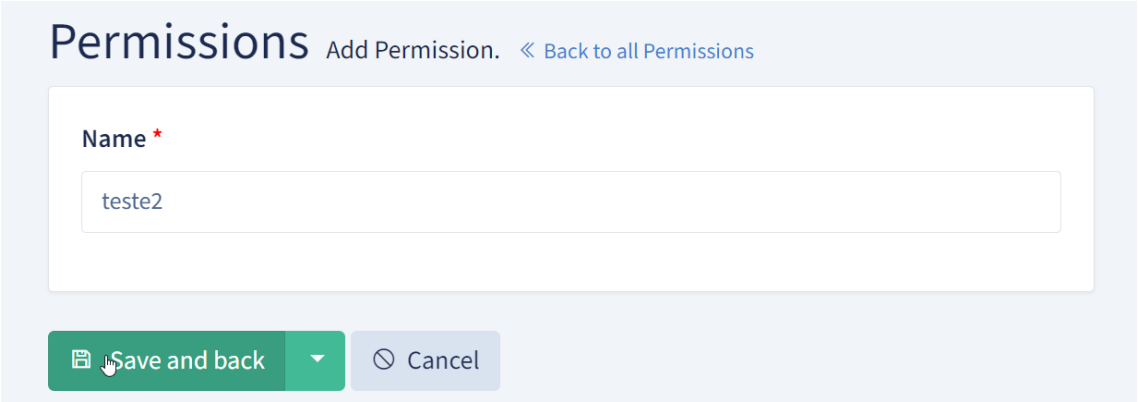
2. O administrador informa o nome da permissão.

Figura 103: Informar o nome da permissão



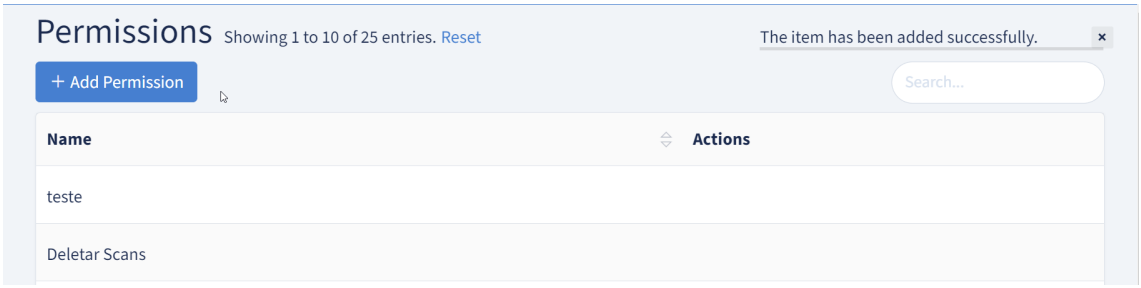
3. O administrador confirma a criação da permissão.

Figura 104: Confirmar a criação da permissão



4. O sistema salva a nova permissão e retorna para a página de listar permissões.

Figura 105: O sistema salva a nova permissão e retorna para a página anterior



Pré-condições

1. O usuário precisa estar logado no sistema Tupã.

Pós-condições

1. Nova permissão é adicionada ao sistema.

Restrições

1. O nome da permissão precisa ter pelo menos 5 caracteres.
2. O usuário precisa ser administrador ou estar habilitado para visualizar permissões no sistema.

Listar Permissão

Esse caso de uso permite que um usuário habilitado possa listar todas as permissões cadastradas no sistema.

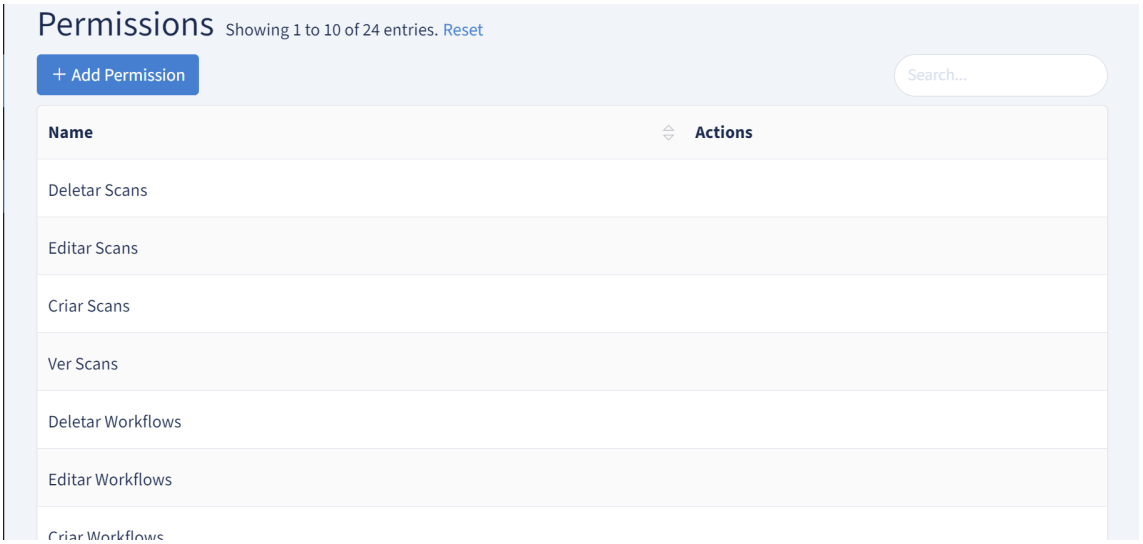
Fluxo Básico

1. O usuário seleciona a opção permissões no menu.

Figura 106: Selecionar a opção "Permissões" no menu

2. O sistema apresenta a lista com todas as permissões cadastradas no sistema.

Figura 107: Lista com todas as permissões cadastradas no sistema



3. O usuário pode selecionar uma permissão específica para visualizar mais informações ou voltar para a página anterior.

Pré-condições

1. O usuário precisa estar autenticado no sistema Tupã.

Pós condições

1. Lista de permissões é apresentada

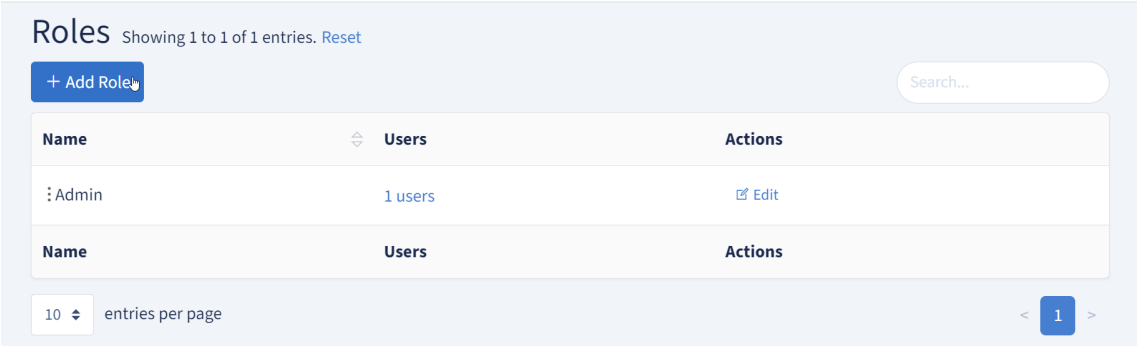
Criar papel

Esse caso de uso permite a criação de um novo papel para um usuário.

Fluxo Básico

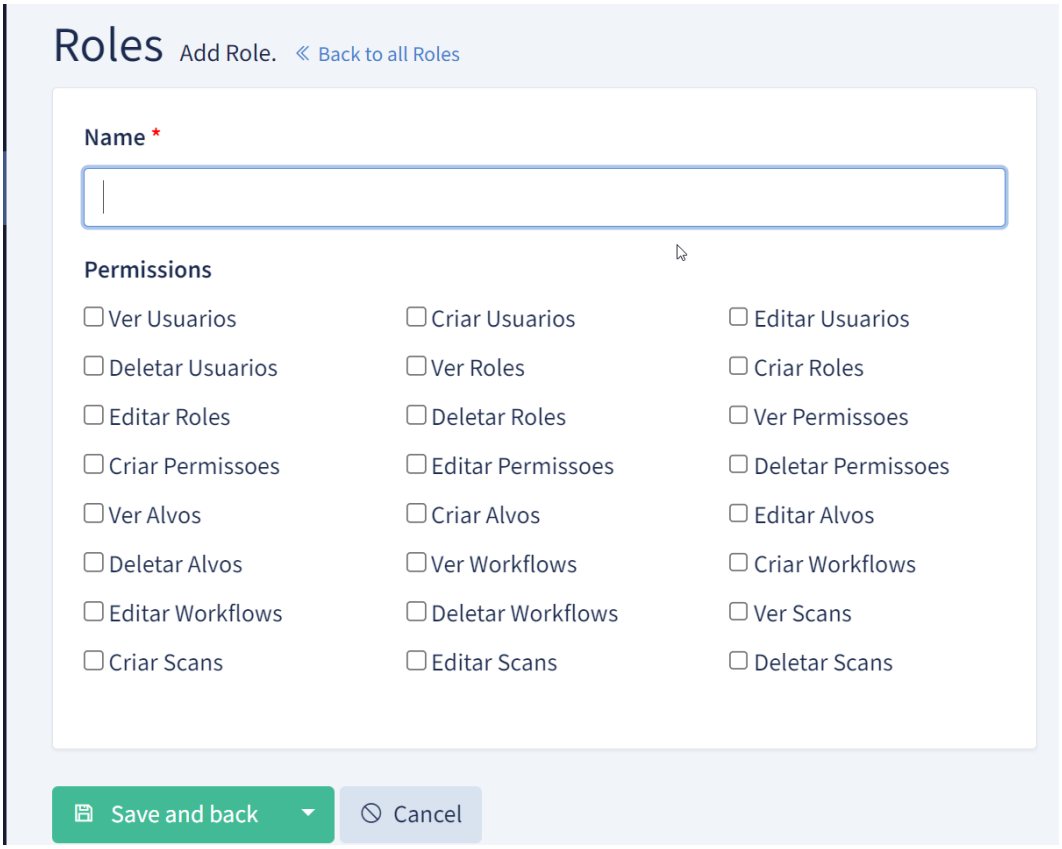
1. O usuário seleciona a opção criar papéis.

Figura 108: Opção "Criar Papéis"



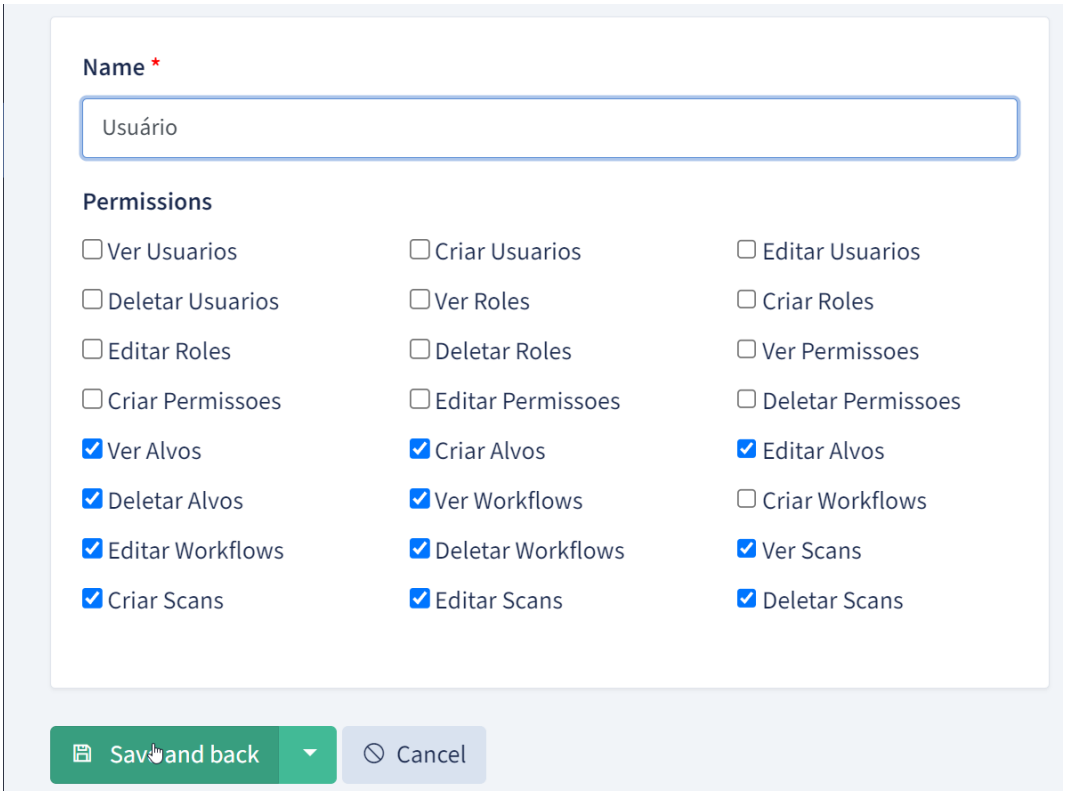
2. O usuário informa o nome e as permissões do papel.

Figura 109: Informar o nome e as permissões do papel



3. O usuário confirma as informações.

Figura 110: Confirmar as informações



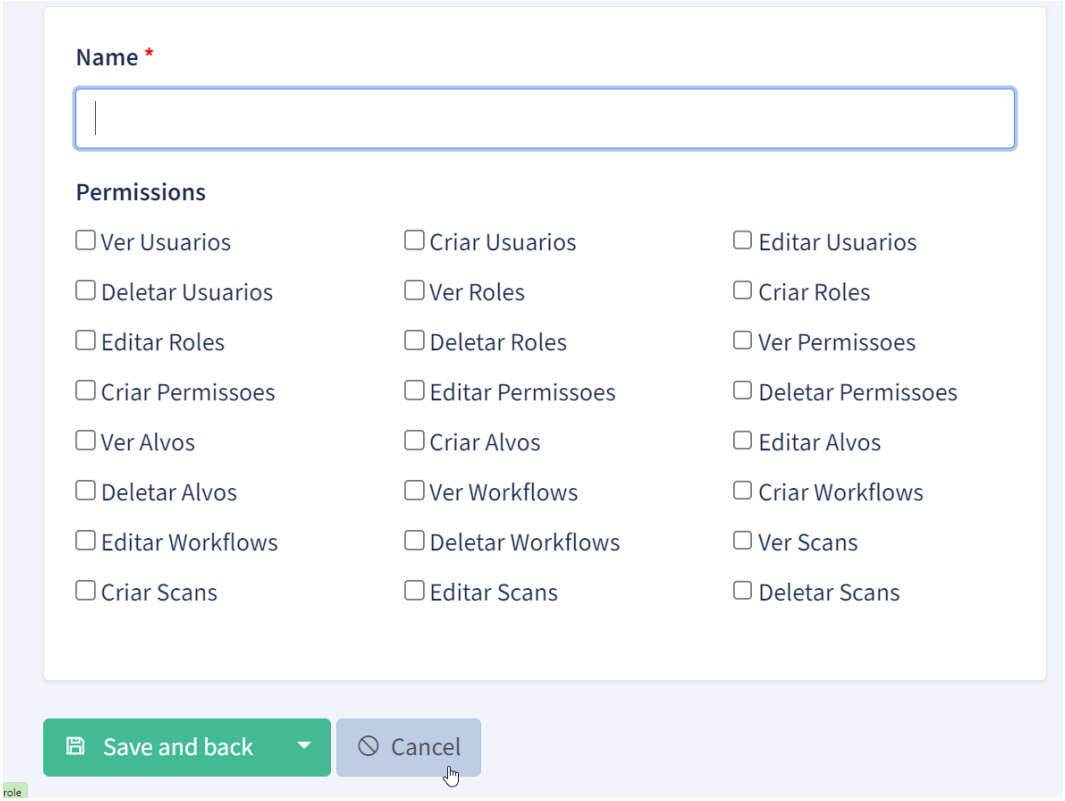
4. O sistema cria um novo papel com as informações fornecidas e o adiciona à lista de papéis existentes.
5. O usuário é redirecionado para a página de listar papéis.

Figura 111: Redirecionamento para a página "Listar Papéis"

Fluxo alternativo

3a) O usuário desiste de criar um novo papel e cancela a operação.

Figura 112: Usuário desisti de criar novo papel e cancela



Pré-condições

1. O usuário precisa executar o caso de uso Listar Papéis.
2. O usuário precisa estar autenticado.

Pós-condições

1. Um novo papel é adicionado ao sistema.

Restrições

1. Não pode haver dois papéis com o mesmo nome.
2. O usuário precisa ter a permissão para criar papel, ou ser um administrador

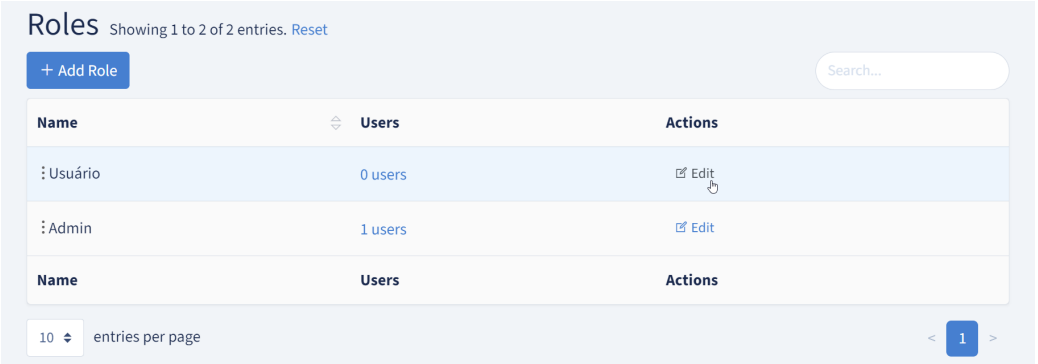
Editar Papel

Esse caso de uso permite a edição de informações de um papel previamente criado.

Fluxo Básico

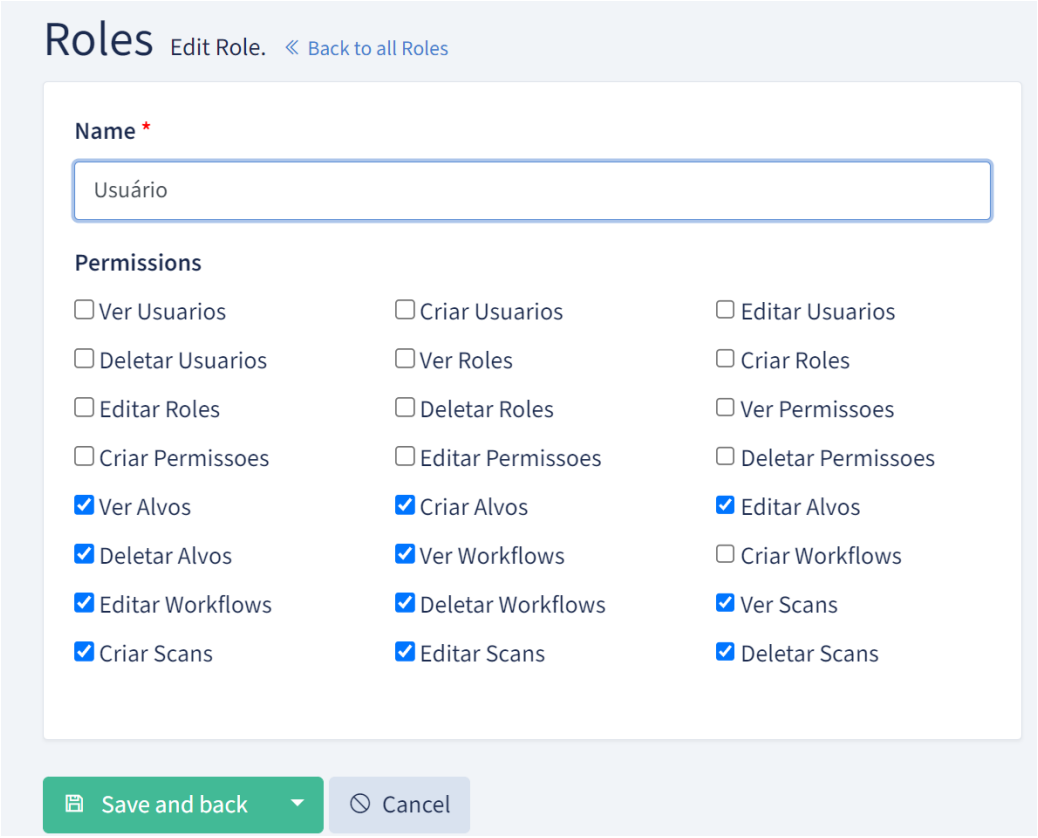
1. Na tela em que os papéis são mostrados o usuário seleciona a opção editar papel.

Figura 113: Selecionar a opção "Editar Papel"



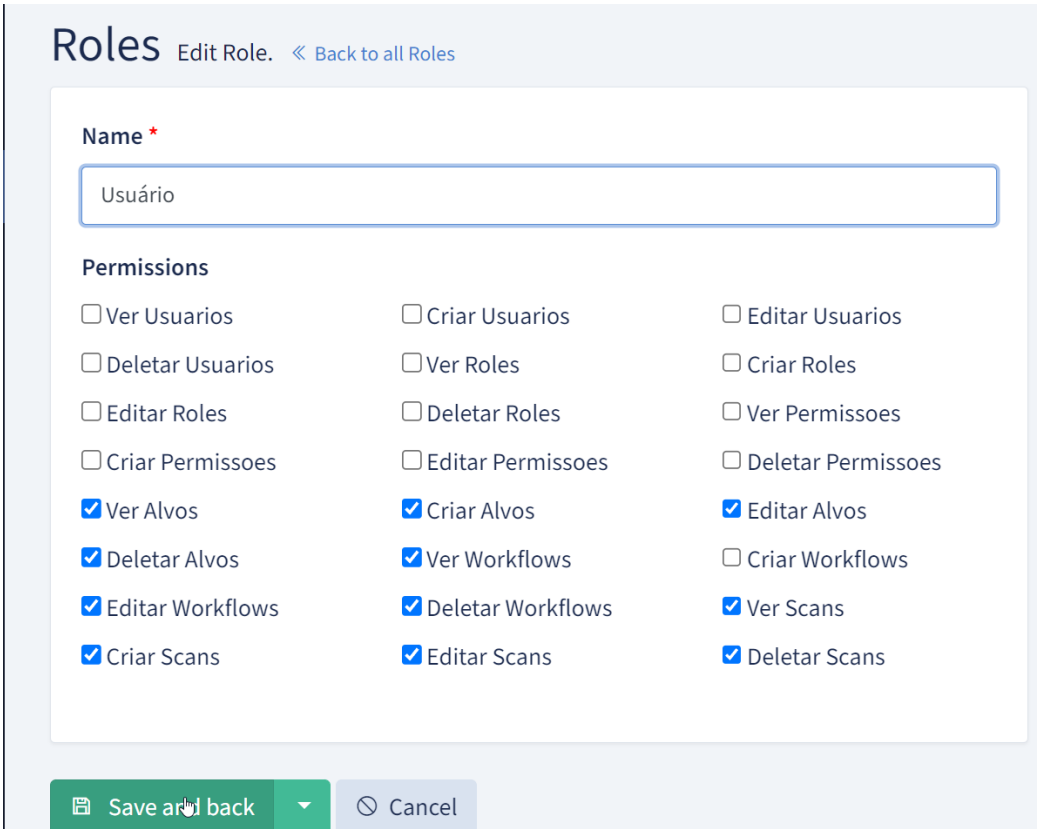
2. O sistema apresenta uma tela com as informações do papel selecionado para edição.

Figura 114: Dados do papel selecionado para edição



3. O usuário edita as informações desejadas e confirma a edição.

Figura 115: Editar e confirmar as edições



4. O sistema exibe uma mensagem informando que as informações foram atualizadas com sucesso e volta para a página de listar papéis.

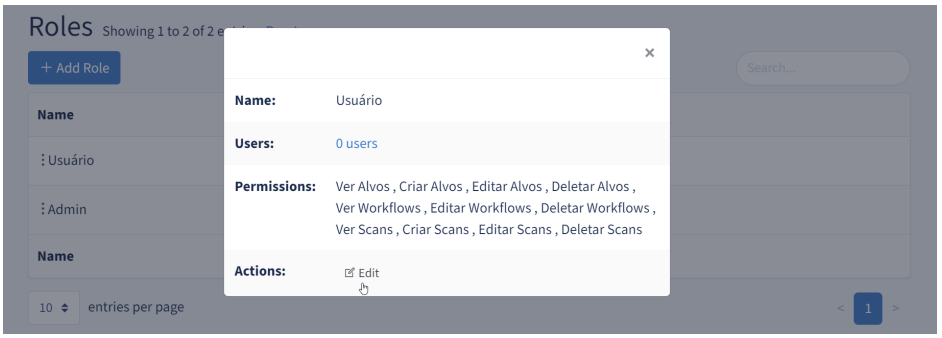
Figura 116: Exibição que as informações foram atualizadas

Fluxo Alternativo

1a)

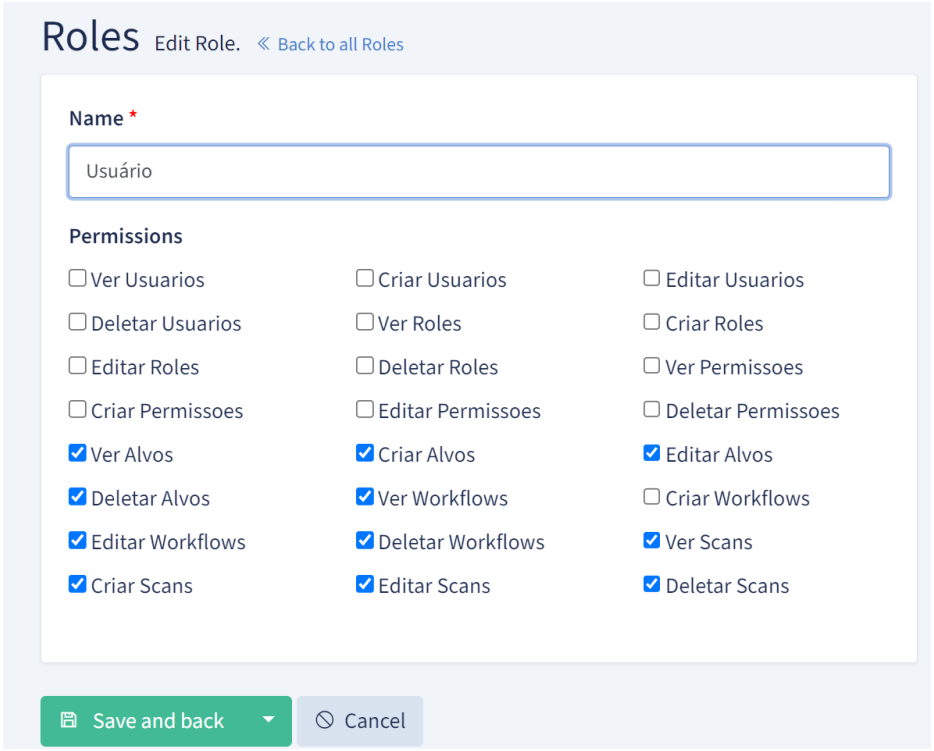
1. Na tela em que o usuário visualiza um papel, ele seleciona a opção editar papel.

Figura 117: Ao visualizar um papel, o usuário seleciona a opção "Editar Papel"



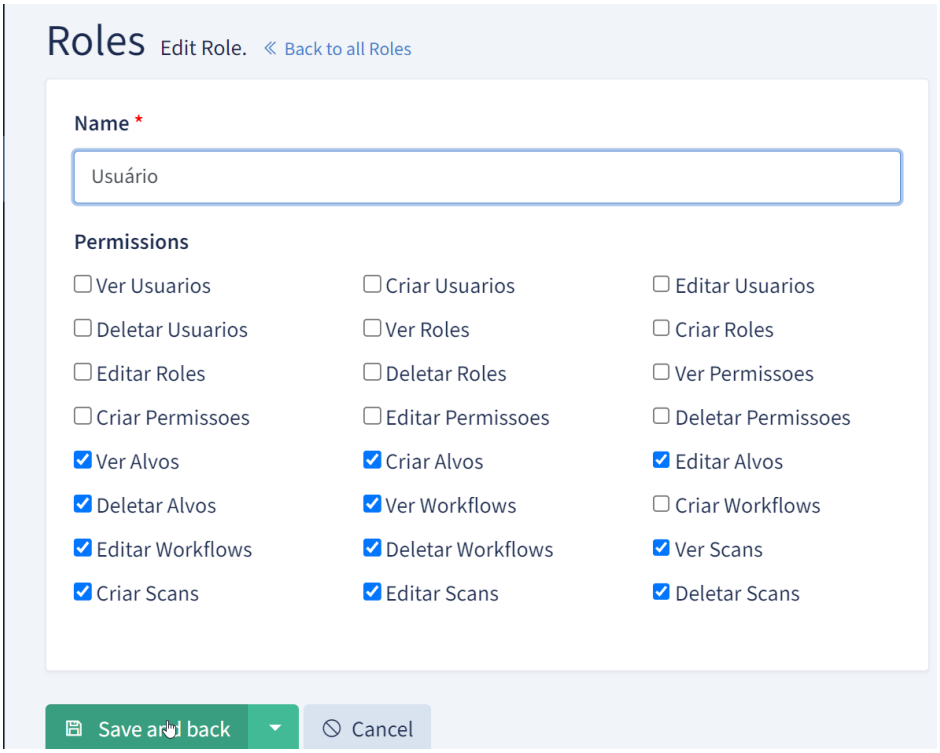
2. O sistema apresenta uma tela com as informações do papel selecionado para edição.

Figura 118: Informações do papel selecionado para editar



3. O usuário edita as informações desejadas e confirma a edição.

Figura 119: Editar as informações e confirmar



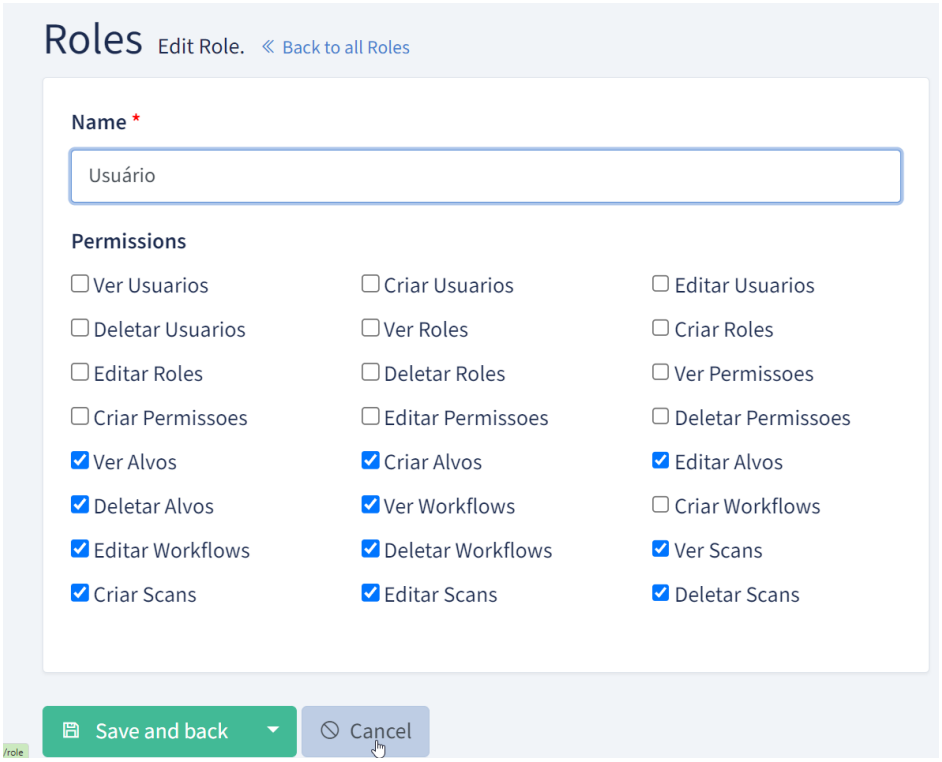
4. O sistema exibe uma mensagem informando que as informações foram atualizadas com sucesso e volta para a página de listar papéis.

Figura 12:0 Exibição das informações que foram atualizadas

3a)

1. O usuário decide não fazer alterações e cancela a operação.

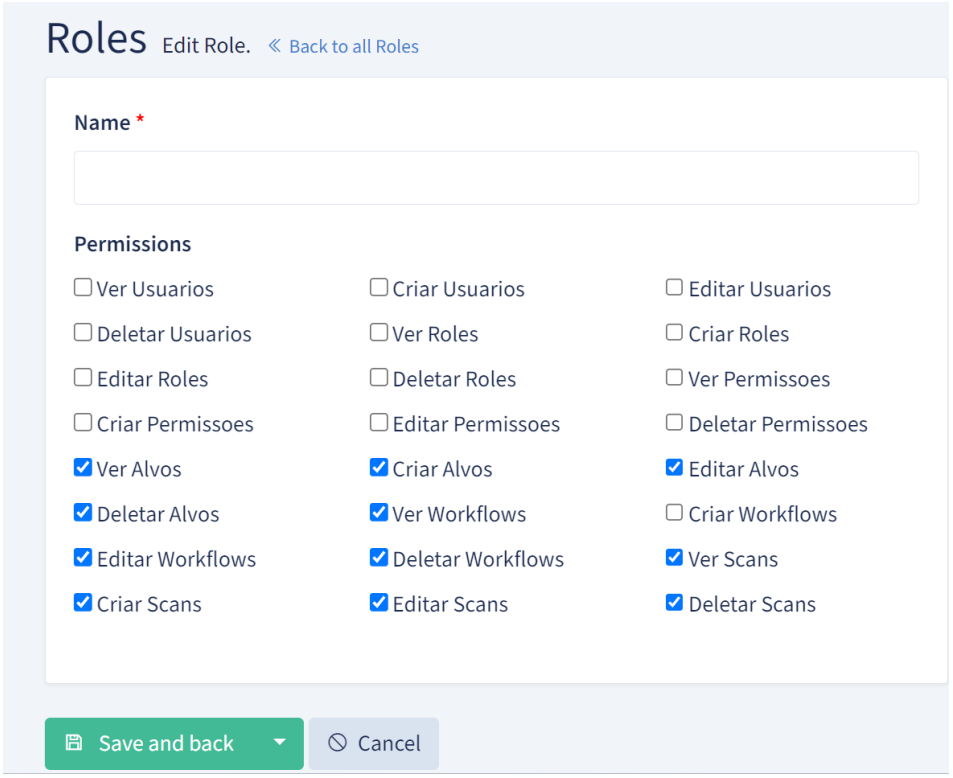
Figura 121: Operação cancelada pelo usuário



3b)

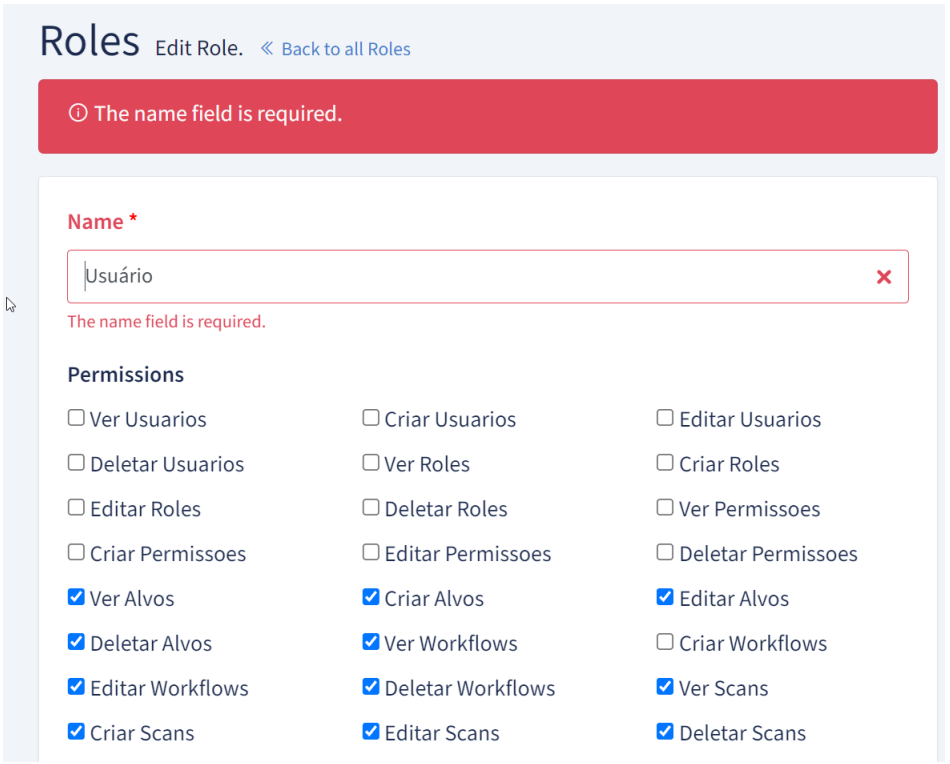
1. O usuário modifica as informações do papel com dados inválidos.

Figura 122: Modificação das informação com dados inválidos pelo usuário



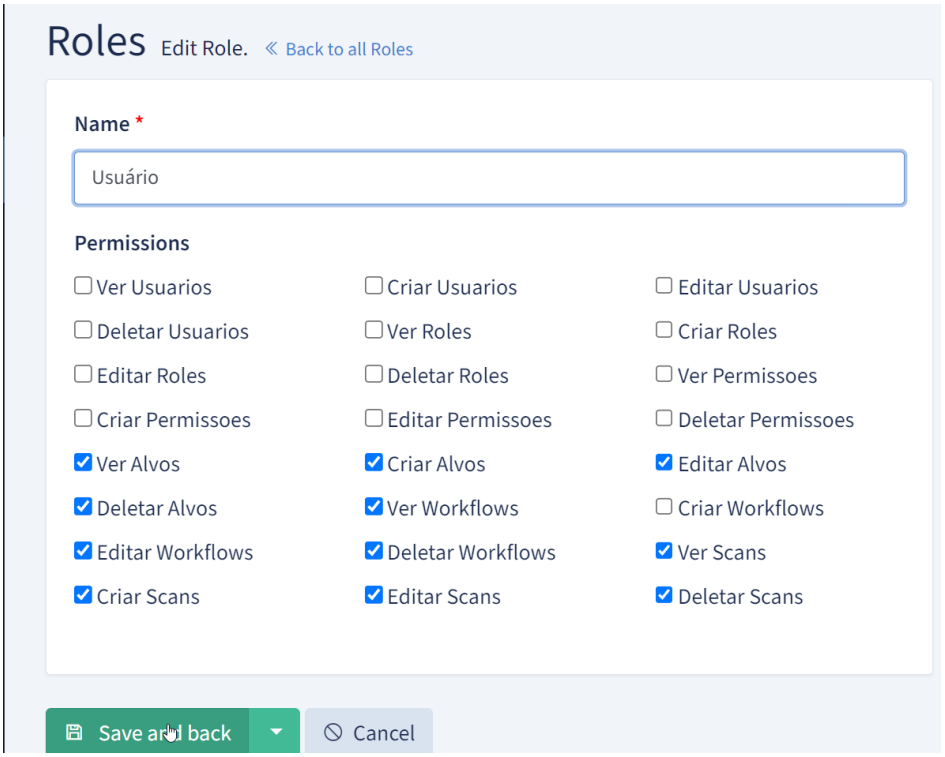
2. O sistema emite uma mensagem de erro informando que os dados são inválidos e solicita ao usuário que insira informações válidas.

Figura 123: Emissão de uma mensagem de erro pelo sistema



3. O usuário realiza a correção dos dados e confirma a alteração novamente.

Figura 124: Realização da correção dos dados e confirmação da alteração



Pré-condições

1. O papel a ser editado já foi criado.
2. O usuário precisa estar autenticado

Pós condições

1. Informações do papel são atualizadas no sistema

Restrições

1. O nome do papel precisa ter pelo menos 5 caracteres
2. O usuário precisa ter a permissão de editar papel ou ser administrador

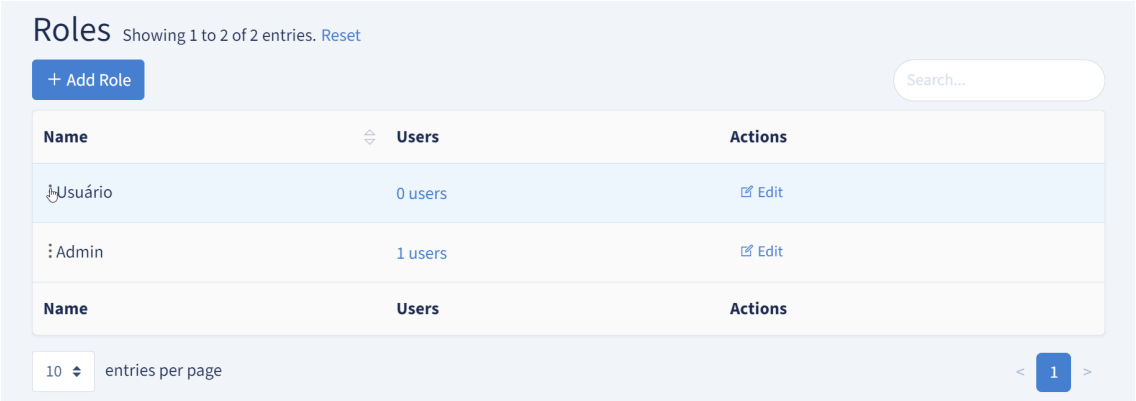
Visualizar Papel

Este caso de uso permite a visualização das informações detalhadas de um papel específico cadastrado no sistema Tupã.

Fluxo Básico

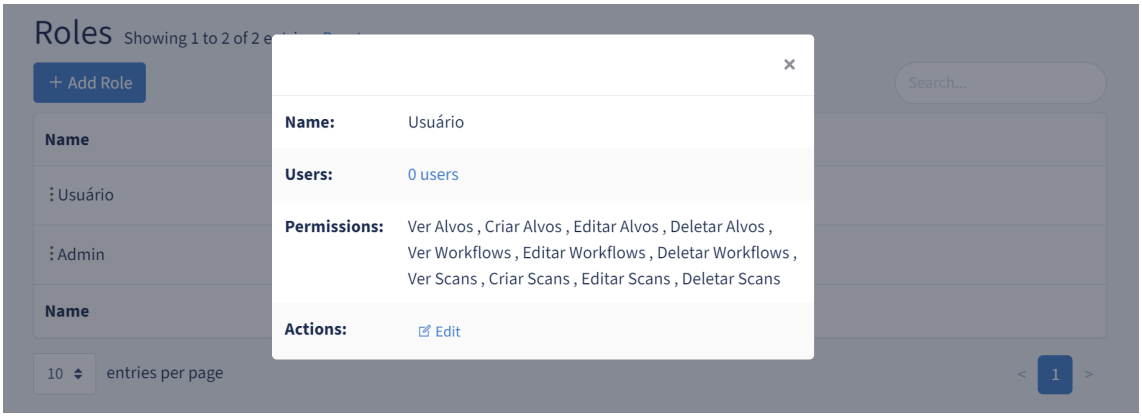
1. O usuário seleciona a opção de visualizar papel na página onde os papéis são listados.

Figura 125: Opção de "Visualizar Papéis"



2. O sistema exibe todas as informações detalhadas do papel selecionado.

Figura 126: Exibição de todas as informações detalhadas



Pré-condições

1. O usuário deve estar autenticado
2. O usuário deve estar na tela de listar papéis.

Pós-condições

O usuário consegue visualizar todas as informações detalhadas do papel selecionado.

Restrições

1. O usuário não pode editar ou excluir o papel nesta página, apenas visualizá-lo.

2. O usuário possui a permissão necessária para visualizar o papel.
3. Caso o papel não exista no sistema Tupã, o usuário deve ser notificado com uma mensagem de erro.

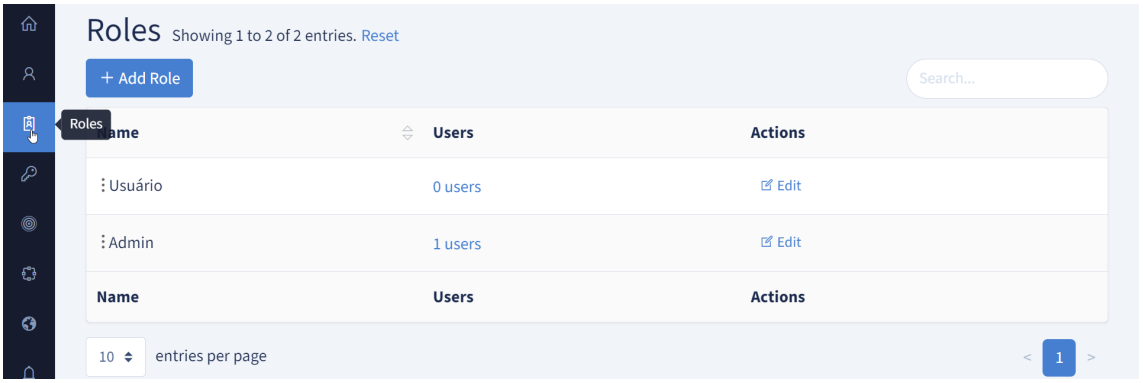
Listar Papéis

Este caso de uso permite que o usuário visualize a lista de todos os papéis cadastrados no sistema Tupã.

Fluxo Básico

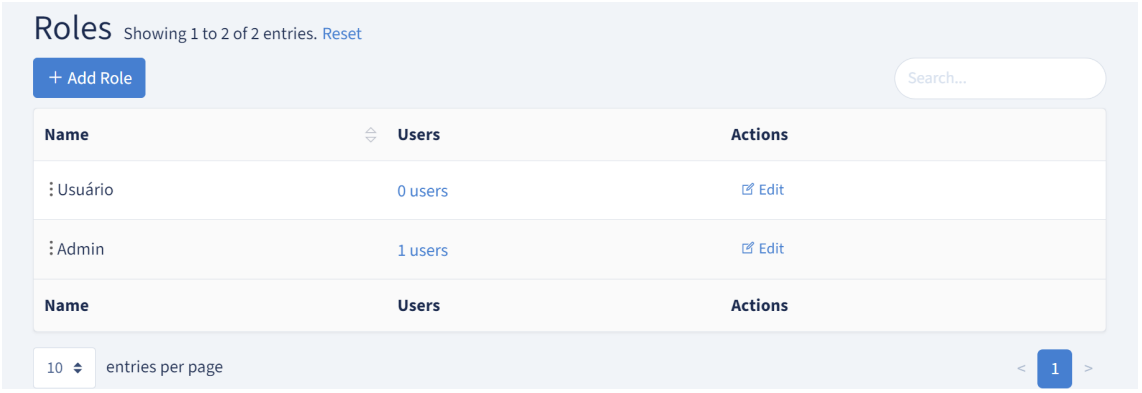
1. O usuário habilitado seleciona a opção papéis no menu.

Figura 127: Selecionar a opção "Papéis" no menu



2. O sistema exibe uma tabela contendo todos os papéis cadastrados, mostrando informações como nome e descrição.

Figura 128: O sistema exibe todos os dados



Pré-condições

1. O usuário precisa estar autenticado no sistema Tupã.

Pós-condições

1. O usuário visualiza a lista de todos os papéis cadastrados no sistema Tupã.

Restrições

1. O usuário precisa ter permissão para listar papéis ou ser administrador do sistema

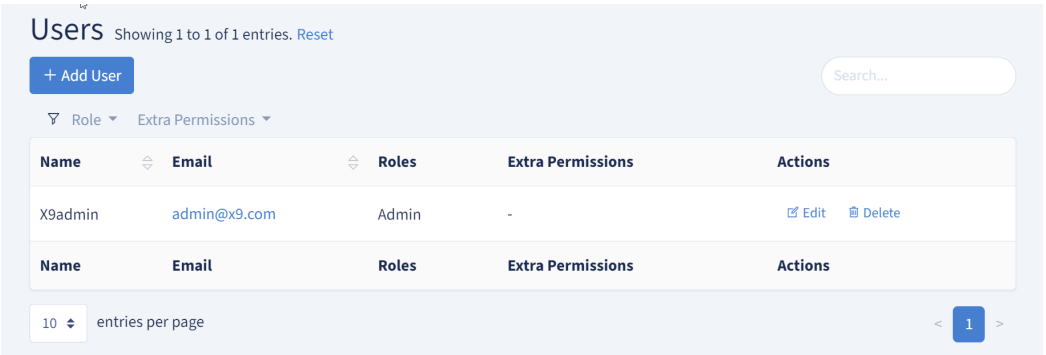
Criar Usuário

Esse caso de uso permite a criação de um novo usuário no sistema.

Fluxo Básico

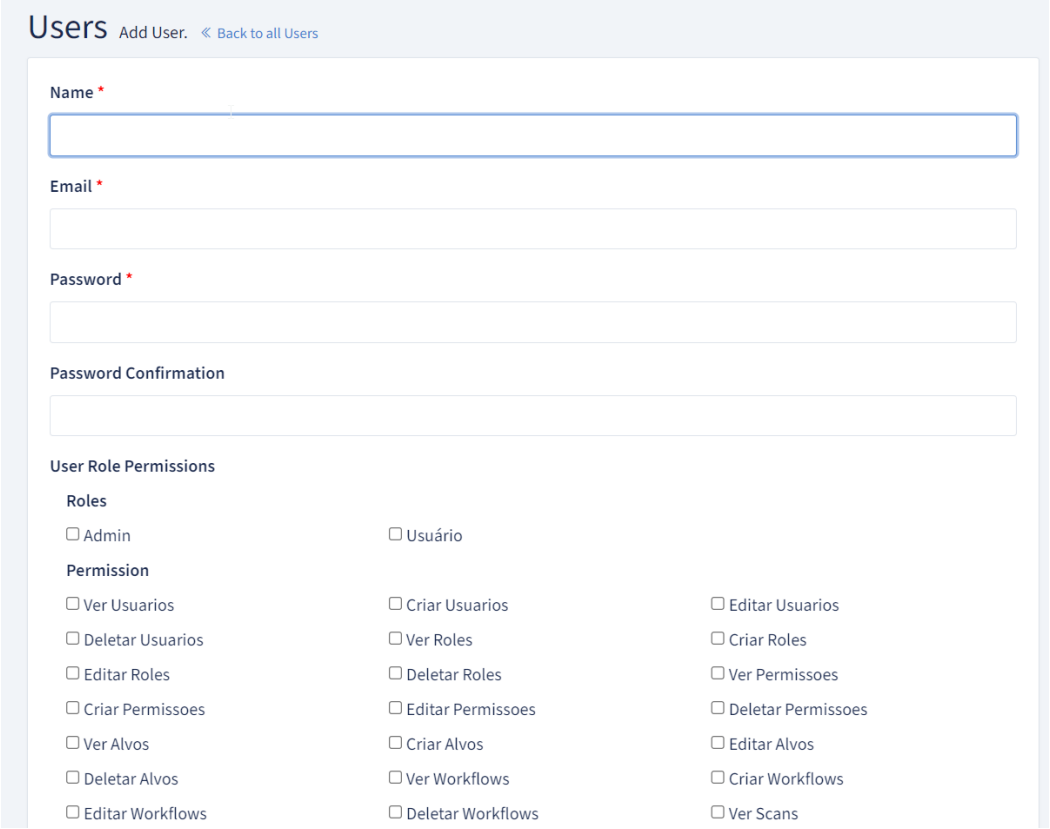
1. O administrador seleciona a opção "Criar Usuário".

Figura 129: Opção "Criar Usuário"



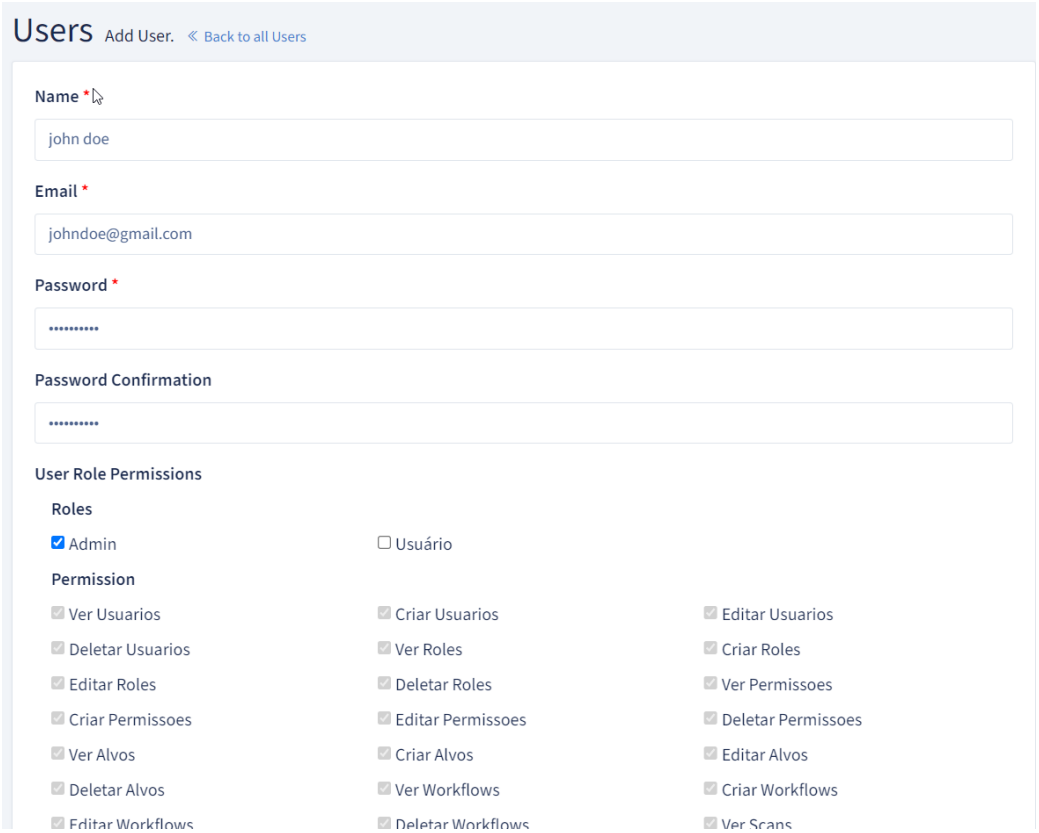
2. O administrador preenche os campos de nome, e-mail, senha e confirmação de senha do novo usuário.

Figura 130: O administrador preenche informações do novo usuário



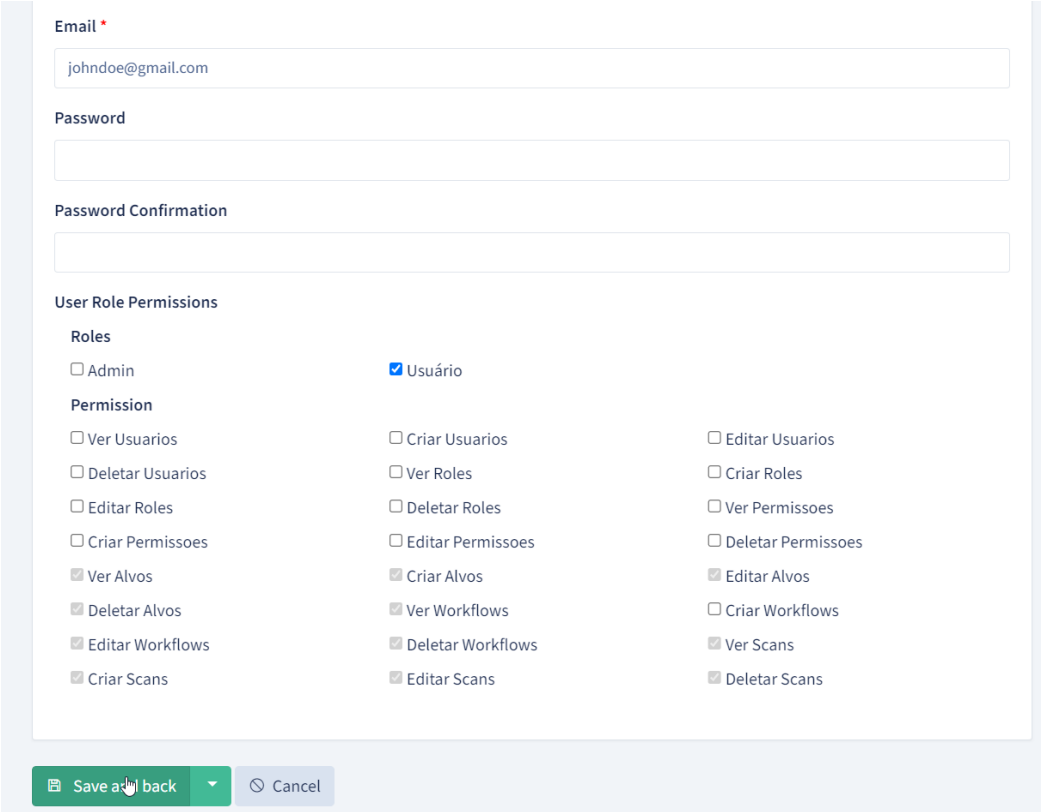
3. O administrador seleciona o papel que será atribuído ao novo usuário.

Figura 131: Selecionar papel que será atribuído ao novo usuário



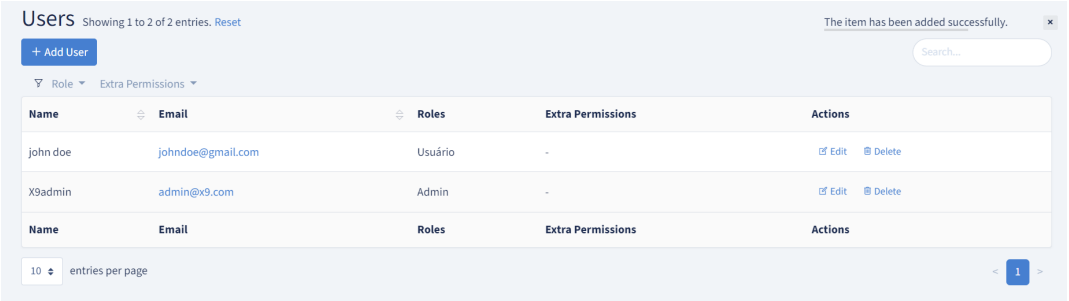
4. O administrador confirma as informações.

Figura 132: Confirmar informações



5. O sistema cria um novo usuário com as informações fornecidas.
6. O administrador é redirecionado para a página de listar usuários.

Figura 133: Redirecionamento para a página de "Listar Usuários"

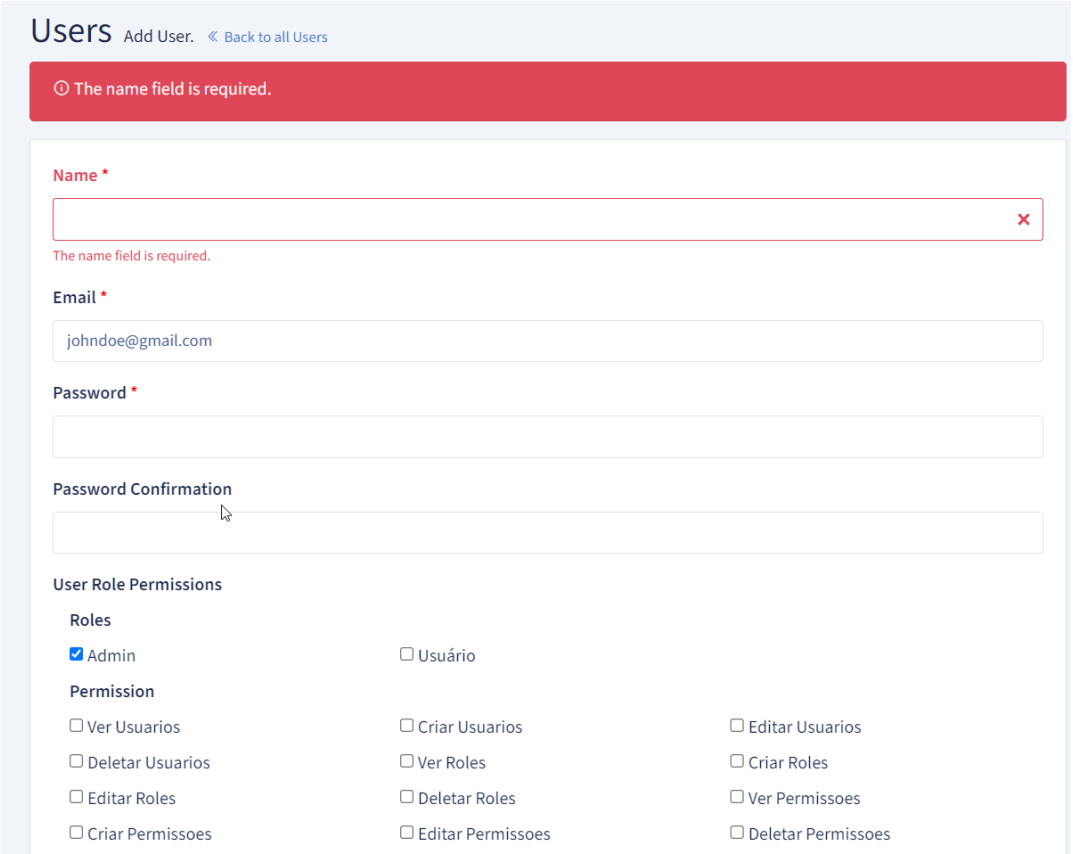


Fluxo Alternativo

4a)

1. mensagem de erro e solicitará que os dados sejam corrigidos.Se o administrador preencher algum campo de forma inválida, o sistema emitirá uma

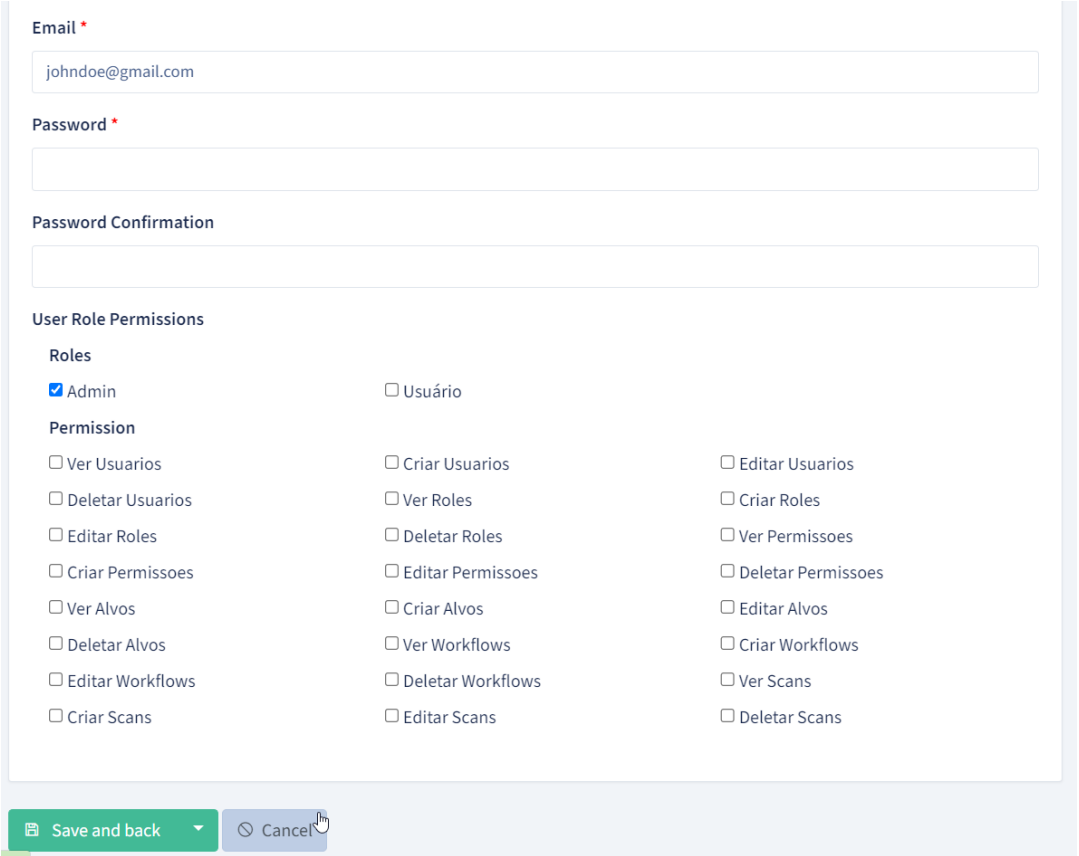
Figura 134: O sistema emite uma mensagem de erro ao preencher algum campo de forma inválida



5a)

1. Se o administrador cancelar a operação, o novo usuário não será criado.

Figura 135: Ao cancelar a operação, o novo usuário não é criado



Pré-condições

1. O usuário precisa estar autenticado como administrador do sistema.

Pós-condições

1. O novo usuário é adicionado ao sistema.

Restrições

1. O e-mail do usuário precisa ser único no sistema.
2. A senha precisa ter pelo menos 8 caracteres e conter pelo menos uma letra maiúscula, uma letra minúscula e um número.

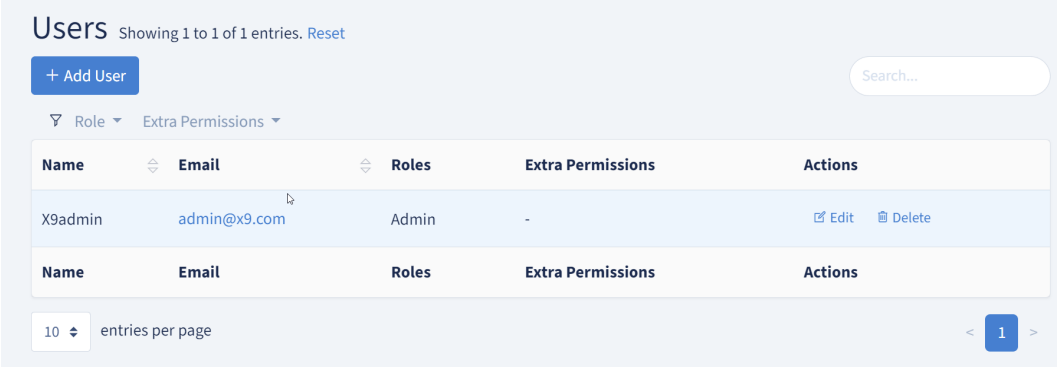
Editar Usuário

Esse caso de uso permite a edição das informações de um usuário previamente cadastrado no sistema.

Fluxo Básico

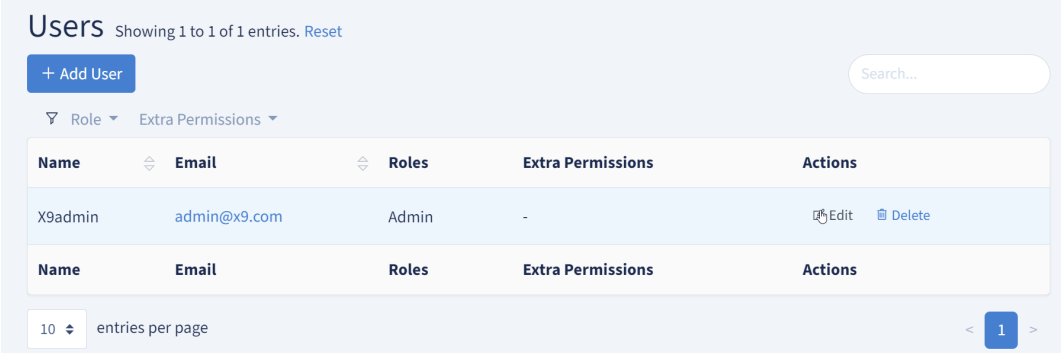
1. O usuário autenticado acessa a página de listar usuários.

Figura 136: Acessar a página de "Listar Usuários"



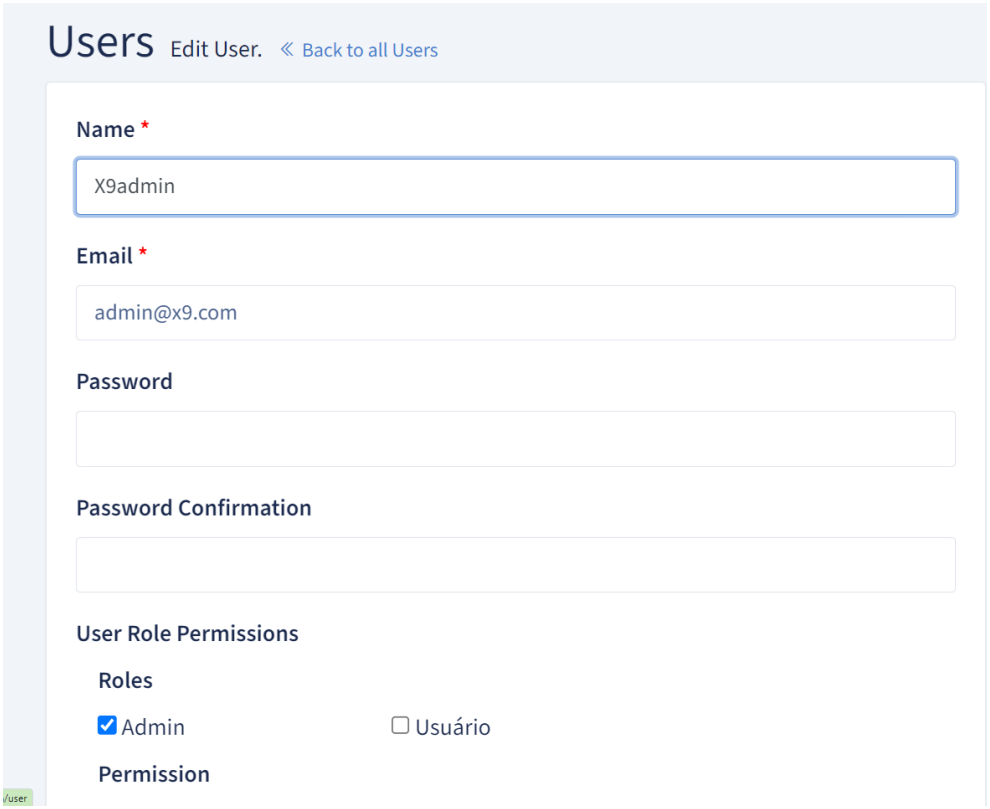
2. O usuário seleciona o usuário que deseja editar.

Figura 137: Selecionar o usuário que deseja editar



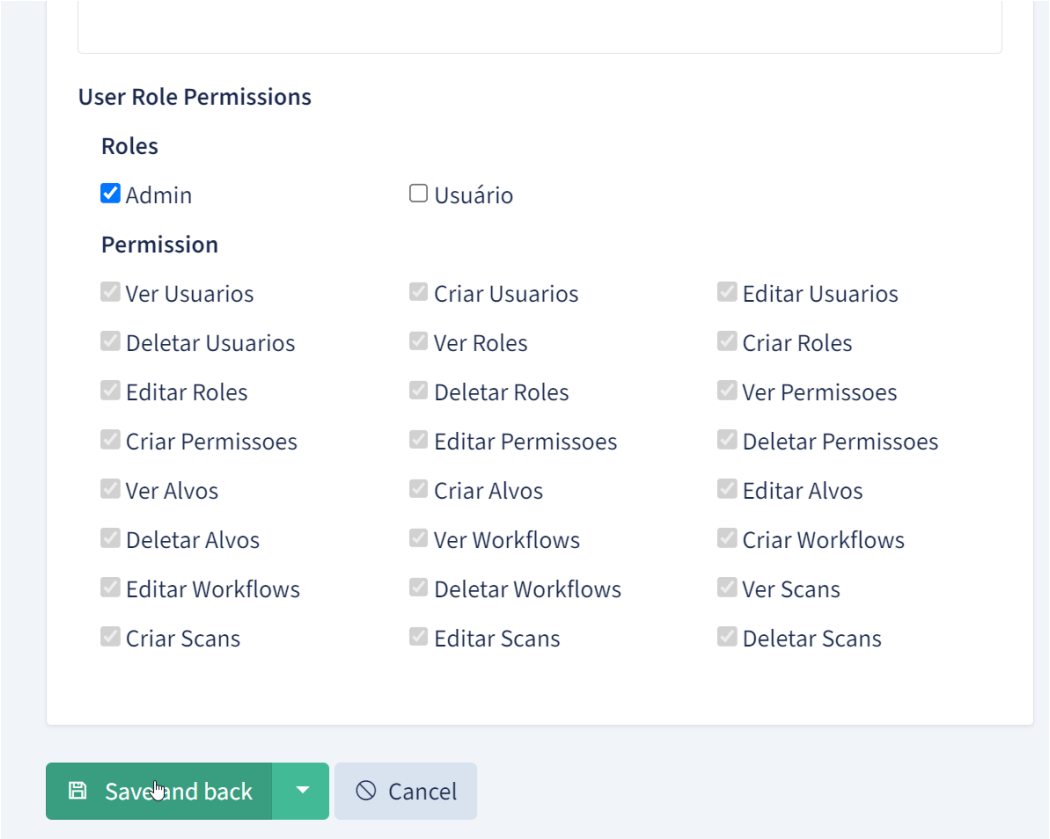
3. O sistema exibe todas as informações do usuário selecionado.

Figura 138: Exibição de todas as informações do usuário selecionado



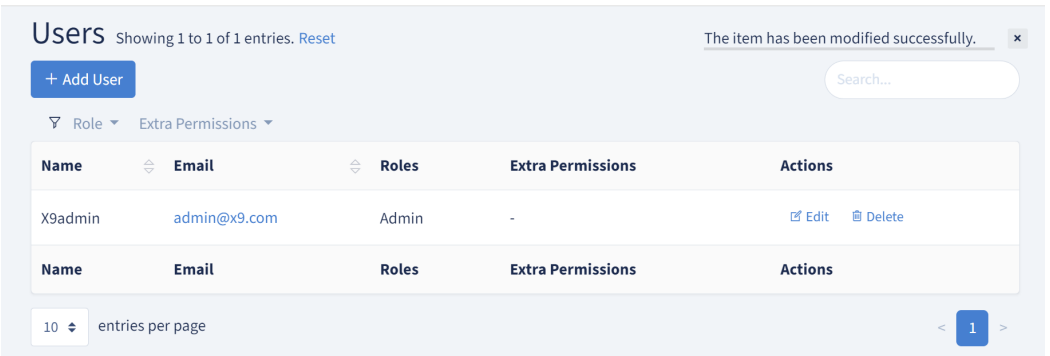
4. O usuário edita as informações desejadas.
5. O usuário confirma a edição das informações.

Figura 139: Confirmar a edição das informações pelo usuário



6. O sistema salva as informações atualizadas do usuário.
7. O usuário é redirecionado para a página de listar usuários.

Figura 140: Redirecionamento para a página "Listar Usuários"

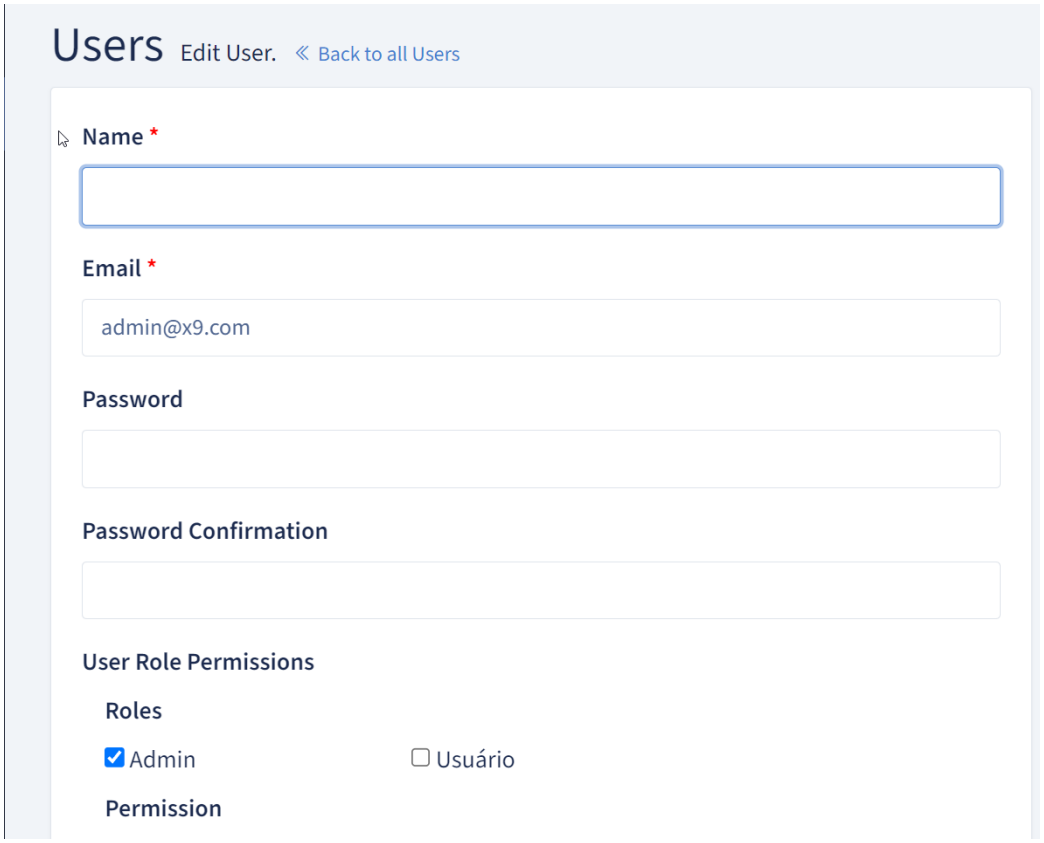


Fluxo Alternativo

4a)

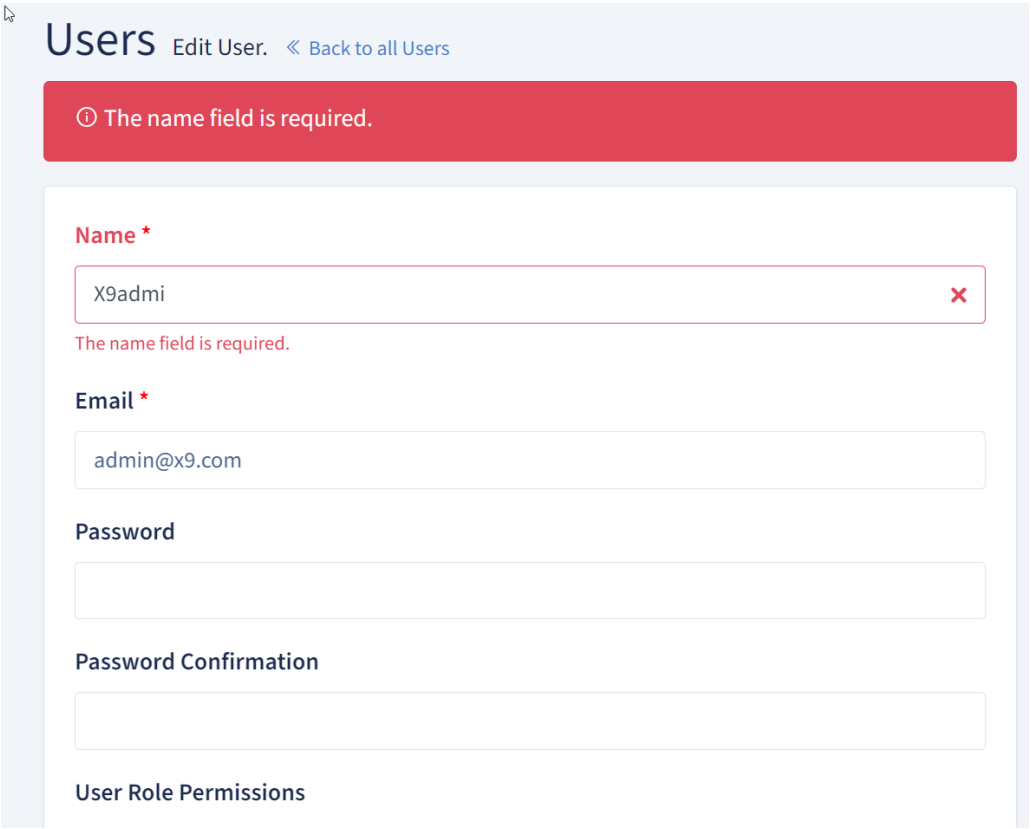
1. O administrador edita as informações do usuário com dados inválidos.

Figura 141: Edição das informações do usuário com dados inválidos



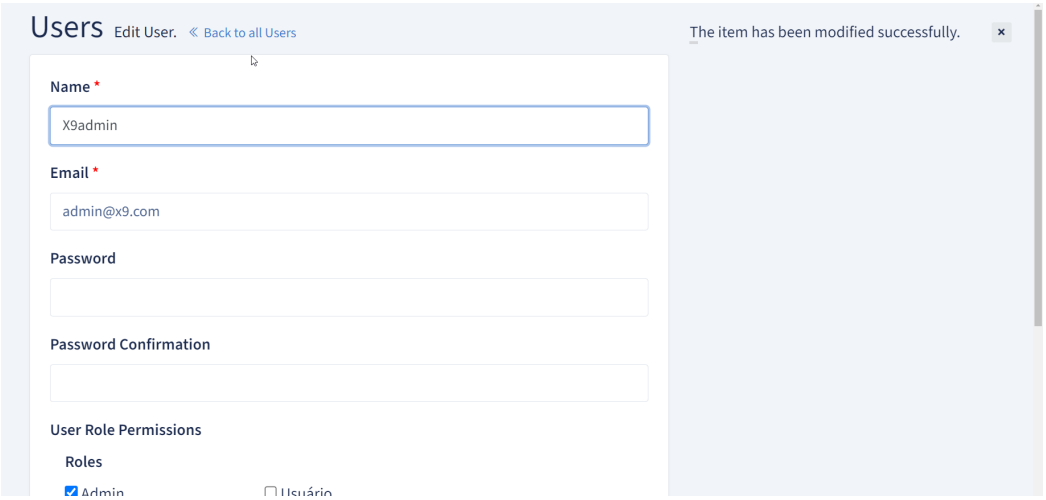
2. O sistema emite uma mensagem de erro informando que os dados são inválidos e solicita ao usuário que insira informações válidas.

Figura 142: O sistema informando que os dados são inválidos e solicitação informações válidas



3. O usuário realiza a correção dos dados e confirma a edição novamente.

Figura 143: Correção de dados de confirmação de nova edição



Pré-condições

1. O usuário deve estar autenticado no sistema.
2. Deve existir pelo menos um usuário cadastrado no sistema.

Pós-condições

As informações do usuário selecionado são atualizadas no sistema.

Restrições

1. O nome de usuário deve ter pelo menos 5 caracteres;
2. O usuário precisa estar cadastrado no sistema para ser editado;
3. O usuário não pode editar o próprio perfil como administrador;
4. O usuário precisa ter permissão para editar usuários ou ser administrador do sistema.

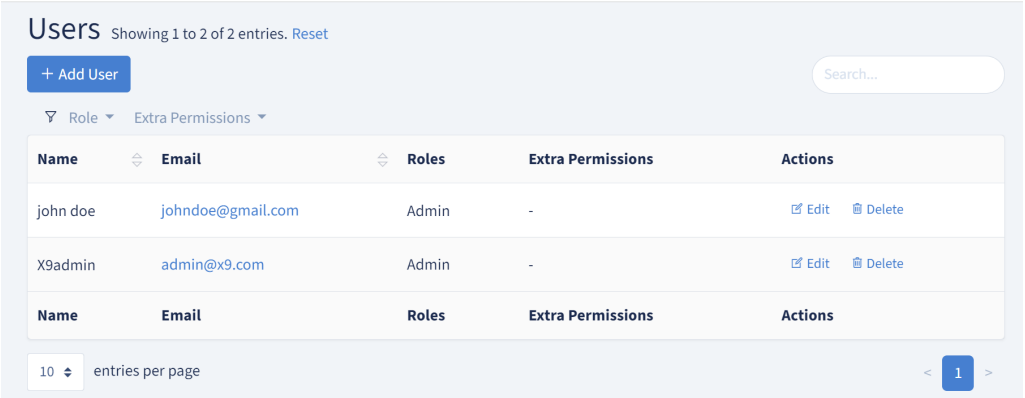
Excluir Usuário

Esse caso de uso permite a exclusão de um usuário previamente cadastrado no sistema.

Fluxo Básico

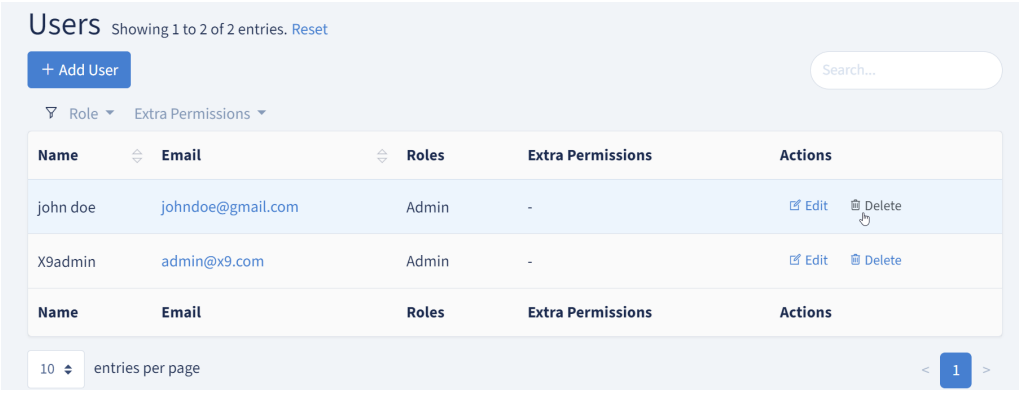
1. O usuário habilitado acessa a lista de usuários.

Figura 144: Acesso a lista de usuários



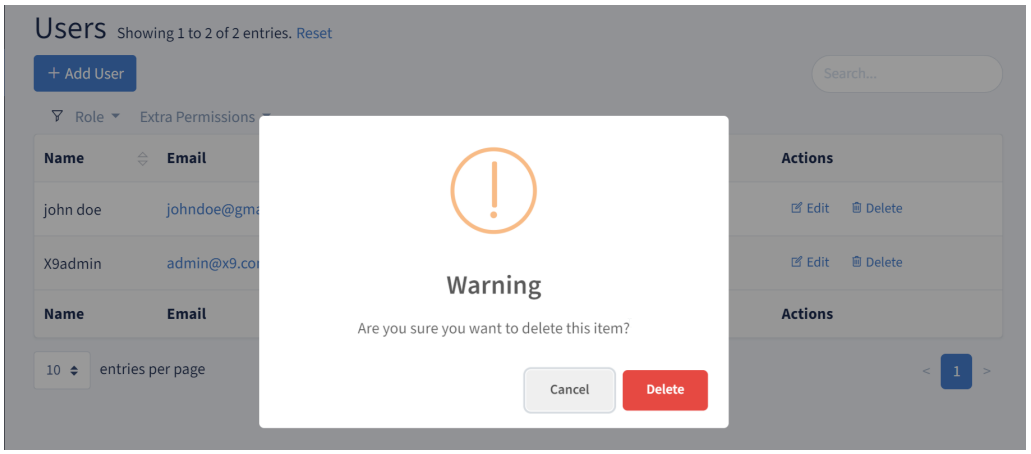
2. O usuário habilitado seleciona o usuário que deseja excluir.

Figura 145: Selecionar usuário que deseja excluir



3. O sistema exibe uma mensagem de confirmação para o usuário.

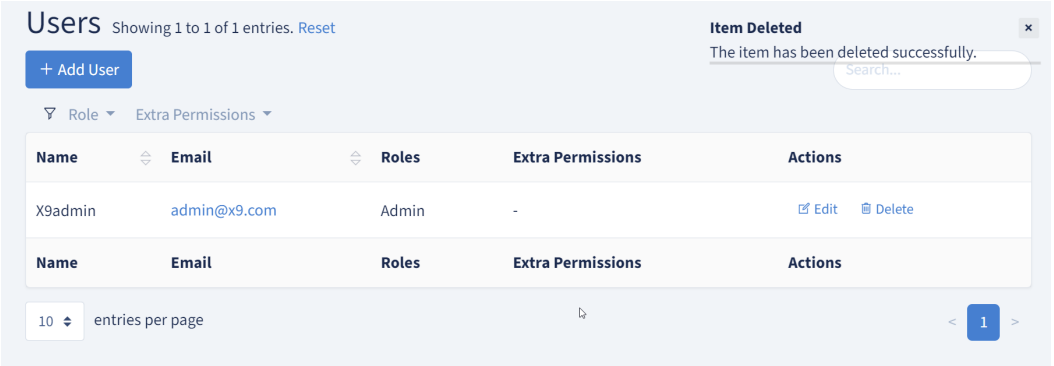
Figura 146: Mensagem de confirmação para o usuário



4. O usuário confirma a exclusão do usuário.

5. O sistema exibe uma mensagem de sucesso informando que o usuário foi excluído com sucesso.

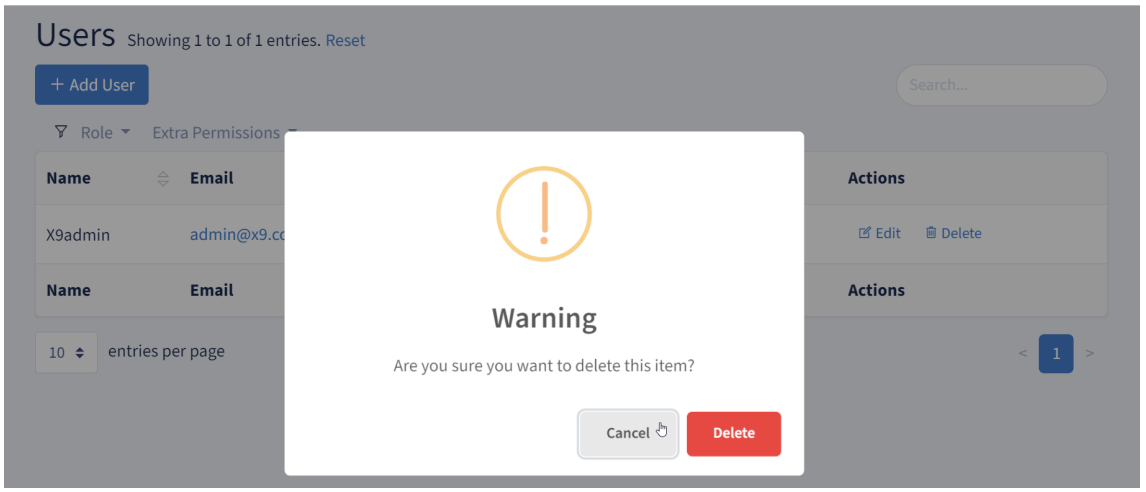
Figura 147: Exibição de uma mensagem que o usuário foi excluído



Fluxo Alternativo

4a) Se o usuário desistir da exclusão, ele pode cancelar a operação e voltar para a página de listar usuários.

Figura *148: Ao desistir da exclusão, o usuário pode cancelar a operação e voltar para a página "Listar Usuários"*



Pré-condições

1. Existe pelo menos um usuário cadastrado no sistema.
2. O usuário precisa estar autenticado no sistema para acessar essa funcionalidade.

Pós-condições

1. O usuário selecionado é excluído do sistema.
2. O sistema volta para a página de listar usuários.

Restrições

1. O usuário só pode excluir usuários que ele mesmo criou.
2. Para excluir um usuário, é necessário deletar todos os dados associados a ele, como scans e relatórios.
3. O usuário tem permissão para excluir usuários ou é administrador.

Listar Usuários

Este caso de uso permite que o usuário visualize a lista de todos os usuários cadastrados no sistema.

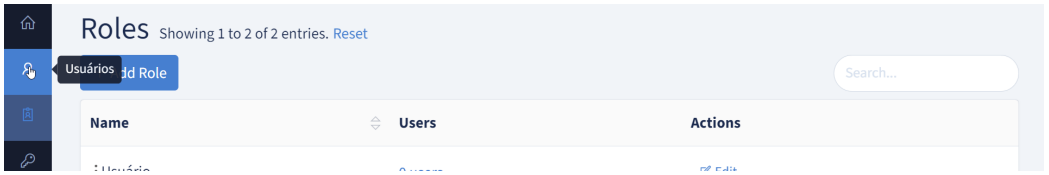
Pré-condições

1. O usuário precisa estar autenticado no sistema.
2. O usuário precisa ter a permissão de visualizar usuários ou ser um administrador do sistema.

Fluxo Básico

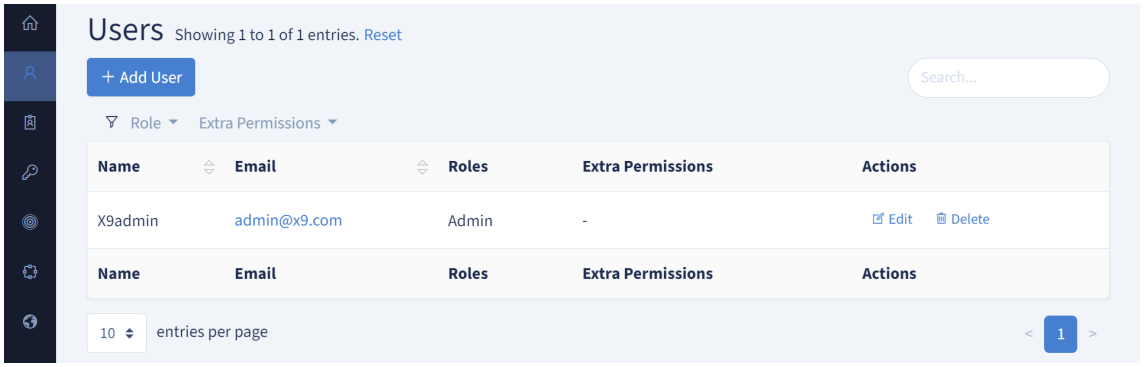
1. O usuário seleciona a opção "Usuários" no menu.

Figura 149: Selecionar a opção "Usuários" no menu



2. O sistema exibe uma tabela contendo todos os usuários cadastrados, mostrando informações como nome, e-mail e permissões atribuídas.

Figura 150: Tabela contendo todas os dados



3. O usuário pode selecionar um usuário específico para editar ou voltar para a página anterior.

Pós-condições

1. O usuário visualiza a lista de todos os usuários cadastrados no sistema.

Apêndice B - Relatório gerado através do Tupã

O relatório pode ser baixado nessa URL:
https://drive.google.com/file/d/1Q1C-AgSYeTptlQi0q2-vYbg25rqRZlsU/view?usp=share_link