UNIVERSIDADE ESTADUAL PAULISTA JÚLIO DE MESQUITA FILHO

FACULDADE DE CIÊNCIAS

BACHARELADO EM SISTEMAS DE INFORMAÇÃO

JULIA FRANCO GRIMALDI

Orientador: Prof. Dr. Kelton Augusto Pontara da Costa

ARGOS-IDS: SISTEMA DE DETECÇÃO E MITIGAÇÃO DE ATAQUES VOLUMÉTRICOS EM REDES DEFINIDAS POR SOFTWARE (SDN)

Bauru, SP

2025

G861a

Grimaldi, Julia
Argos-IDS : Sitema de Detecção e Mitigação de Ataques Volumétricos em Redes Definidas por Software (SDN) / Julia Grimaldi. -- Bauru, 2025
27 p. : il., tabs.

Trabalho de conclusão de curso (Bacharelado - Sistemas de Informação) - Universidade Estadual Paulista (UNESP), Faculdade de Ciências, Bauru
Orientador: Kelton Augusto Pontara da Costa

1. Sistema de Detecção de Intrusão (IDS). 2. Redes Definidas por Software (SDN). 3. Aprendizado de Máquina. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Dados fornecidos pelo autor(a).

Resumo

Este trabalho apresenta o desenvolvimento do ARGOS-IDS, um sistema interativo de detecção e mitigação de ataques volumétricos voltado para redes definidas por software (SDN). A proposta busca oferecer uma solução de baixo custo e fácil implementação, especialmente voltada para pequenas e médias organizações que não dispõem de equipes especializadas em segurança da informação. O sistema combina o uso de expressões regulares como pré-filtro com técnicas de aprendizado de máquina para a identificação de tráfego malicioso, atuando de forma automatizada por meio de um controlador SDN. A interface web local proporciona ao usuário uma visualização clara da rede, além de permitir a gestão de regras e modos de mitigação de maneira simples e centralizada. Ao tornar tecnologias avançadas mais acessíveis, o ARGOS-IDS contribui para a democratização da cibersegurança, promovendo a proteção de redes mesmo em ambientes com infraestrutura limitada.

Palavras-chave: Sistema de Detecção de Intrusão (IDS). Redes Definidas por Software (SDN). Aprendizado de Máquina.

Abstract

This work presents the development of ARGOS-IDS, an interactive system for the detection and mitigation of volumetric attacks in Software-Defined Networks (SDN). The proposed solution aims to offer a low-cost, easy-to-implement tool, especially designed for small and medium-sized organizations that lack specialized cybersecurity teams. The system combines regular expression-based pre-filtering with machine learning techniques to identify malicious traffic, acting automatically through an SDN controller. The local web interface provides users with a clear view of network activity and allows centralized management of mitigation rules and modes. By making advanced technologies more accessible, ARGOS-IDS contributes to the democratization of cybersecurity, enabling effective network protection even in environments with limited infrastructure.

Keywords: Intrusion Detection System (IDS), Software Defined Network (SDN), Machine Learning.

Lista de Ilustrações

Lista de Tabelas

Sumário

1 Introdução

Nos últimos anos, a transformação digital acelerou a migração de processos, serviços e interações para o ambiente virtual. Segundo a União Internacional de Telecomunicações (UIT), em 2023, mais de 5,4 bilhões de pessoas ao redor do mundo estavam conectadas à internet, correspondendo a aproximadamente 67% da população mundial (UIT. . . , 2023). No Brasil, os números também são significativos, com mais de 84% da população tendo acesso à internet, segundo o IBGE (NERY, 2023). Nesse contexto, estabelecer uma presença digital deixou de ser simplesmente uma alternativa e se tornou uma exigência para organizações de todos os portes e setores, que dependem da conectividade para garantir a continuidade e crescimento no mercado.

Embora traga muitos benefícios, essa expansão também apresentou desafios significativos para as empresas, especialmente no campo da *cibersegurança*. Com cada vez mais sistemas conectados, os planos de ataque também se tornam mais complexos, deixando as redes corporativas mais vulneráveis a ameaças digitais constantes. Entre os perigos, se destacam os ataques distribuídos de negação de serviços (DDoS), que surgem como uma preocupação crucial. Esses ataques agem através de um grande volume de tráfego, resultando na exaustão de recursos e, consequentemente, na negação de serviços a tráfegos legítimos (CHOU; GROVES, 2018).

Uma das formas mais comuns desses ataques, são os ataques volumétricos, uma forma de ataque DDoS, que sobrecarregam a rede com grandes quantidades de pacotes de dados, consumindo a largura de banda disponível e prejudicando o processamento de solicitações legítimas (NIN, 2023). Com a evolução das técnicas de ataque, a frequência dessas ameaças cresce junto com a necessidade de soluções eficazes para proteger redes e sistemas.

É nesse contexo que se insere o ARGOS-IDS, um sistema inovador de detecção e mitigação de ataques volumétricos em redes definidas por software (SDN), integrando expressões regulares (Regex) e técnicas de *machine learning*. Por meio de um modelo de aprendizado de máquina, o sistema identifica padrões de tráfego associados a ataques volumétricos. Ao envolver técnicas de *machine learning* na arquitetura SDN, busca-se melhorar a detecção, analisando o tráfego para identificar comportamentos maliciosos e ajustar políticas de encaminhamento. A proposta também inclui a implementação do Regex como pré-filtro, visando identificar padrões específicos no tráfego e redirecioná-lo, essa estratégia aumenta a eficiência da solução ao reduzir o volume de dados e melhorar a capacidade de detecção do sistema.

Além disso, o ARGOS-IDS é uma plataforma interativa voltada para organizações que não contam com equipes especializadas em *cibersegurança*. Além da detecção e mitigação dos ataques, ele oferece funcionalidades como monitoramento, alertas e gerenciamento de políticas de defesa de forma simples e centralizada. O objetivo é oferecer uma solução robusta de baixo custo, que permita ampliar a proteção e gerenciamento de redes de organizações, sem exigir

infraestrutura complexa ou conhecimento técnico avançado.

1.1 Detalhamento do Problema

O principal desafio na defesa contra ataques volumétricos não está apenas em sua intensidade, mas na limitação das infraestruturas tradicionais em identificar, de forma precisa e em tempo real, o que é tráfego malicioso e o que são apenas variações legítimas da rede. Essa complexidade torna a detecção baseada em limites fixos inadequada. Pequenos desvios criados por um invasor são suficientes para contornar mecanismos tradicionais, que ou falham em responder quando necessário, resultando em falsos negativos, ou reagem de forma excessiva, interrompendo tráfego legítimo e causando falsos positivos.

A situação se agrava ainda mais quando levamos em conta a realidade de pequenas e médias empresas (PMEs). Segundo o relatório *SMB Cybersecurity Report* da Microsoft, 94% das PMEs reconhecem que a segurança digital é fundamental para suas operações, mas, a maioria delas não possui as ferramentas, profissionais ou infraestrutura necessária para manter suas defesas sempre atualizadas (SMALL…, 2024). O estudo também mostra que um em cada três pequenos e médios negócios sofreu um ataque no último ano e que o custo médio de um incidente ultrapassa US$ 250 mil, podendo chegar a US$ 7 milhões nos casos mais graves. A vulnerabilidade dessas empresas não está só na tecnologia, mas também na maneira como elas operam. Muitas enfrentam dificuldades como a falta de especialistas internos, treinamentos regulares que não acontecem com frequência, uso crescente de dispositivos pessoais pelos funcionários e dificuldades para manter políticas de segurança consistentes. Além disso, há uma percepção distorcida sobre o risco em muitos desses ambientes. Algumas empresas que sofreram ataque acreditam que já não serão mais atacadas, enquanto outras acham que, por serem pequenas demais, não são alvos. Essa visão equivocada aumenta ainda mais a chance de serem vulneráveis a ataques digitais.

Diante desse cenário, se torna claro que organizações que mais precisam de mecanismos de defesa são justamente as que menos possuem recursos para implementá-los. Soluções que se baseiam apenas em listas de controle, *firewalls* tradicionais ou regras de detecção estáticas tendem a falhar porque não conseguem acompanhar a variabilidade do tráfego real. Por outro lado, abordagens baseadas apenas em *machine learning*, apesar de poderosas, enfrentam limitações práticas como o alto consumo de recursos computacionais, latência na análise de grandes volumes de pacotes, e necessidade de equipes capacitadas para ajustar modelos e interpretar seus resultados. Para PMEs, essas restrições tornam sistemas puramente inacessíveis e, na maioria das vezes, inviáveis.

É nesse ponto que a necessidade de uma abordagem híbrida se apresenta como resposta direta ao problema. O uso de expressões regulares no pré-processamento do tráfego permite identificar rapidamente padrões explícitos e conhecidos, como origens suspeitas, protocolos, portas ou formatos de fluxos relacionados a comportamentos anômalos. Trata-se de uma camada

leve e de alta velocidade e que não exige *hardware* sofisticado. Essa filtragem inicial diminui a quantidade de inspeções profundas, aliviando a carga do sistema e permitindo que técnicas mais custosas, como algoritmos de *machine learning*, funcionem de forma eficiente sobre um subconjunto menor e mais relevante de fluxos. Além disso, as expressões regulares (Regex) proporcionam um nível de controle direto ao administrador, permitindo ajustes imediatos diante de incidentes emergenciais, algo fundamental para empresas que não contam com SOCs (Centro de Operações de Segurança) dedicados.

Por sua vez, modelos de *machine learning* complementam essa primeira etapa ao atuar sobre características do tráfego que não podem ser formalmente definidas por regras. Em ataques modernos, grande parte dos comportamentos maliciosos não apresentam um padrão claro, mas se manifestam por anomalias comportamentais. A detecção automática desses padrões só é possível por meio de análise preditiva, capaz de identificar desvios que escapam da inspeção humana e de regras estáticas. Dessa forma, o *machine learning* não substitui o pré-filtro, mas corrige suas limitações, permitindo que o sistema identifique tentativas de evasão ou tráfego malicioso que se disfarça como atividade legítima.

1.2 Estrutura

Este trabalho está organizado da seguinte forma: o Capítulo 2 apresenta as soluções existentes no mercado e na literatura, destacando suas limitações; o Capítulo 3 descreve a arquitetura e os módulos do sistema proposto; o Capítulo 4 aborda as tecnologias utilizadas no desenvolvimento da solução; o Capítulo 5 trata da validação do sistema e dos testes realizados; o Capítulo 6 apresenta as conclusões obtidas e sugestões para melhorias e trabalhos futuros.

2 Soluções Existentes

A segurança de redes tem se tornado uma prioridade para organizações de todos os portes, especialmente diante do aumento constante dos ataques DDoS. Com isso, diversas soluções foram desenvolvidas ao longo dos últimos anos para detectar e mitigar esse tipo de ameaça. Essas soluções variam em complexidade, preço e aplicabilidade, sendo muitas dessas voltadas apenas para ambientes corporativos de grande porte. Neste capítulo, serão apresentadas algumas das principais soluções existentes, com foco especial em alternativas que servem de referência ou contraste para o desenvolvimento do ARGOS-IDS.

2.1 Solução baseada em *Machine Learning*

Diversas pesquisas acadêmicas têm explorado o uso de técnicas de aprendizado de máquina para a detecção de ataques DDoS em Redes Definidas por *Software* (SDN). Essas soluções visam melhorar a segurança de redes SDN, aproveitando a flexibilidade e o controle centralizado dessas infraestruturas. Técnicas como *Random Forest, Support Vector Machine (SVM), K-Nearest Neighbors (KNN)* e redes neurais têm sido aplicadas para classificar tráfego de rede e identificar padrões anômalos que indicam a presença de ataques DDoS.

Por exemplo, o trabalho "*DDoS Detection in SDN using Machine Learning Techniques*"propõe um sistema de detecção de ataques DDoS em redes definidas por *software* (SDN) utilizando técnicas de aprendizado de máquina. A abordagem avalia diferentes métodos de seleção de atributos combinados a algoritmos de classificação, como *Random Forest, SVM e KNN,* alcançando até 99,97% de acurácia. Embora tecnicamente eficaz, a solução é voltada à experimentação acadêmica, não oferecendo interface interativa nem funcionalidades de gestão em tempo real que permitam seu uso direto por pequenas empresas (NADEEM HOCK GUANGOH, 2021).

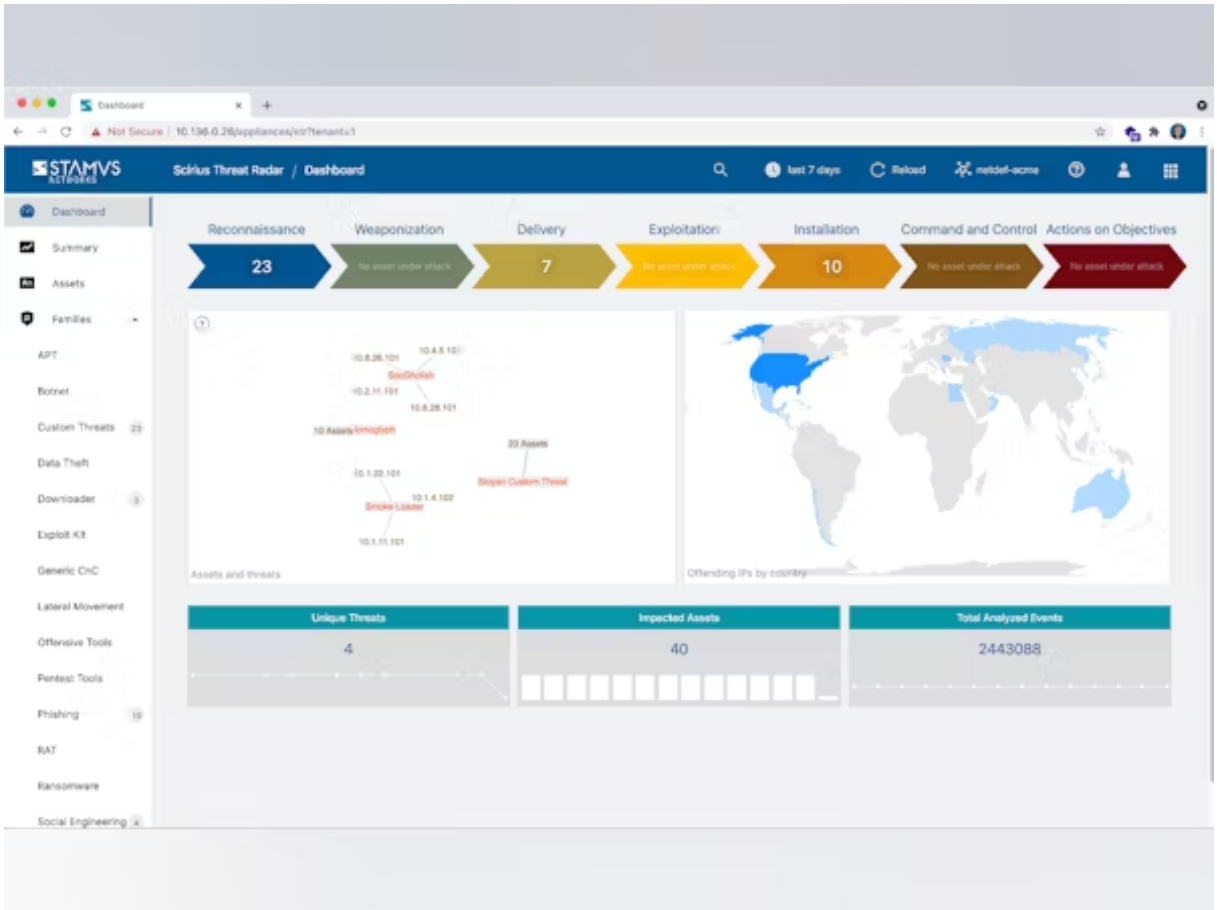
2.2 Cloudflare DDoS Protection

A Cloudflare é uma das provedoras de serviços de proteção contra DDoS mais conhecidas do mercado. Sua solução atua de forma distribuída em nuvem, mitigando ataques volumétricos antes que eles atinjam a infraestrutura do cliente. O sistema protege não apenas *websites*, mas também APIs e aplicações, com suporte a detecção automática e resposta imediata. Embora ofereça um plano gratuito com proteção básica, os recursos avançados estão disponíveis apenas em planos pagos, sendo mais adequados para médias e grandes empresas. Outro ponto é que, por ser baseada em nuvem, a solução depende de redirecionamento de tráfego, o que pode não ser ideal em todos os cenários (DISTRIBUTED. . . , 2025).

2.3 Stamus Security Platform (SSP)

A Stamus oferece uma solução moderna de *Network Detection and Response* (NDR) (Figura 1) que incorpora funcionalidades de IDS sem os desafios tradicionais associados a essas tecnologias. A plataforma utiliza métodos avançados, como *machine learning* e análise comportamental, e se destaca pela visibilidade ampla da rede e pela eficiência no tratamento de alertas. Um diferencial importante é o modelo de precificação: em vez de cobrar por IP ou usuário, o preço é baseado no número e velocidade dos *links* monitorados, com valores a partir de US$ 6.000 por ano por *probe*. Apesar de ser mais acessível que outras soluções corporativas, ainda está fora do orçamento da maioria das pequenas empresas e exige conhecimento para implantação e operação (STAMUS. . . , 2025).

Figura 1 – Stamus Security Platform



Fonte: Adaptado de Stamus Networks. Disponível em: <https://www.stamus-networks.com/>. Acesso em: 20 jun. 2025.

2.4 Considerações Finais

As soluções analisadas demonstram a maturidade e a diversidade de abordagens no mercado de cibersegurança. No entanto, também evidenciam uma lacuna relevante: a ausência de ferramentas voltadas especificamente para pequenas e médias organizações que não contam com

Tabela 1 – Comparação entre Soluções de Detecção e Mitigação de DDoS

Critério	**SDN com ML (Acadêmico)**	**Cloudflare DDoS Protection**	**Stamus Security Platform (SSP)**	**ARGOS-IDS (Proposto)**
Tipo de Solução	Pesquisa acadêmica experimental	Serviço distribuído em nuvem	Plataforma NDR comercial	Solução local open-source
Foco em SDN	Sim	Não	Parcial	Sim
Interface Interativa	Não possui	Sim (dashboard web)	Sim (dashboard corporativo)	Sim (plataforma local e web)
Complexidade de Implantação	Alta	Baixa (serviço externo)	Média a alta	Baixa
Custo Estimado	N/A (experimental)	Médio a alto	Alto (a partir de US$ 6.000/ano)	Baixo
Detecção com Machine Learning	Sim	Sim (heurísticas + ML)	Sim	Sim
Pré-filtro Regex	Não	Não	Não	Sim
Público-Alvo	Pesquisa e testes acadêmicos	Grandes empresas	Médias e grandes empresas	Pequenas e médias empresas

Fonte: Elaborado pela autora com base em (NADEEM HOCK GUANGOH, 2021; DISTRIBUTED..., 2025; STAMUS..., 2025).

equipes técnicas especializadas ou orçamentos elevados. Soluções como a proposta acadêmica baseada em SDN e *machine learning* demonstram viabilidade técnica, mas não apresentam aplicabilidade direta em ambientes reais. Outras como a Stamus e Cloudflare oferecem proteção eficaz, mas são voltadas para cenários mais complexos e com alto custo de entrada.

O ARGOS-IDS surge como uma proposta inovadora justamente por preencher esse espaço negligenciado. Trata-se de uma solução de código aberto, de baixo custo e com foco na usabilidade, permitindo que organizações sem estrutura dedicada de segurança possam implementar proteção contra ataques volumétricos. Seu diferencial está na combinação de detecção inteligente baseada em *machine learning* com pré-processamento por expressões regulares e gestão via plataforma interativa. Além disso, o ARGOS-IDS oferece uma experiência acessível através de *dashboards*, alertas em tempo real, monitoramento de rede e gerenciamento de regras, tudo centralizado em uma interface simples. Assim, o sistema não apenas identifica e mitiga ataques, mas também empodera organizações a manterem sua infraestrutura segura com autonomia e baixo investimento. Na Tabela 1 apresenta-se uma comparações geral entre as soluções apresentadas no capítulo.

3 Metodologia

O sistema ARGOS-IDS foi projetado como uma solução completa e acessível para detecção e mitigação de tráfego malicioso em ambientes baseados em Redes Definidas por *Software* (SDN). A metodologia de desenvolvimento combinou prototipagem incremental e validação prática em laboratório, com foco em quatro eixos principais: coleta de fluxos em um controlador SDN, pré-filtragem com regras configuráveis, classificação automática por aprendizado de máquina e mitigação integrada ao plano de controle da rede. Todo o núcleo do sistema foi implementado em *Python*, utilizando o controlador *Ryu*, bibliotecas para ciência de dados (*Pandas, NumPy, Scikit-learn*), um banco de dados leve em *SQLite* e uma *API web* desenvolvida com *FastAPI* para integração com o painel de monitoramento em *HTML, CSS e JavaScript.*

3.1 Sistema de Detecção de Intrusão (IDS) – Processamento, Classificação e Mitigação

O componente de detecção do ARGOS-IDS foi implementado como um aplicativo *Ryu* (`ControllerAPI`), que atua diretamente sobre o plano de controle da rede SDN. Esse módulo consulta periodicamente, via *API REST* do *Ryu* (`/stats/flow` e `/stats/switches`), as estatísticas de fluxo mantidas pelos *switches OpenFlow* e, a partir desses dados, executa todo o pipeline de coleta, pré-processamento, classificação e mitigação.

A coleta de dados ocorre em ciclos contínuos, por meio da função *_monitor*, que identifica os *switches* ativos e chama o método `collect_and_store_stats` para cada *switch*. Para cada fluxo retornado pela *API* do *Ryu*, o sistema extrai informações como endereço IP de origem e destino, portas de transporte, protocolo, contagem de pacotes e *bytes*, tempo de duração do fluxo e *timeouts*. A partir dessas informações, são calculadas métricas derivadas, como taxa de pacotes por segundo e taxa de *bytes* por segundo, garantindo que o modelo de aprendizado de máquina opere sobre atributos mais expressivos do ponto de vista de detecção.

Os fluxos coletados são persistidos em dois níveis. Primeiro, são armazenados em um arquivo CSV (*traffic_predict.csv*) destinado ao *pipeline* de inferência. Em paralelo, o sistema utiliza um banco *SQLite* local (*traffic.db*) para registrar fluxos já classificados, regras de pré-filtro, contatos de alerta e bloqueios ativos. Essa combinação de CSV e banco relacional simplifica a análise posterior dos dados, além de evitar dependência de infraestruturas externas de banco de dados.

Antes que o tráfego chegue ao classificador, o ARGOS-IDS aplica um pré-filtro baseado em regras configuráveis. Essas regras são armazenadas na tabela *filter_rules* do *SQLite* e carregadas pelo controlador por meio do método *_load_filter_rules*. Cada regra pode

especificar padrões de endereço IP de origem e destino (que são convertidos em expressões regulares), protocolo, portas de origem e destino, além de limites de volume como número máximo de *bytes*, pacotes, taxa de pacotes por segundo (*PPS*) e taxa de *bytes* por segundo (*BPS*). A função `_flow_matches_rule` avalia, para cada fluxo, se esses critérios são satisfeitos, permitindo que o sistema aplique ações de bloqueio ou apenas de alerta sem envolver, necessariamente, o modelo de aprendizado de máquina.

O modelo de classificação utilizado pelo ARGOS-IDS foi treinado a partir de um conjunto de dados público e utilizado para estudos de detecção de tráfego malicioso em redes SDN. O dataset (ALAWADI, 2025), contém fluxos de rede extraídos em ambiente SDN com etiquetas de tráfego benigno e malicioso.

A etapa de classificação automática utiliza um modelo de *k-nearest neighbors (k-NN)* previamente treinado. Esse modelo é armazenado em disco como um *bundle* em formato `pickle`, contendo o classificador, o *scaler* aplicado às variáveis numéricas e a lista de atributos esperados. No momento da inicialização, o método `_load_models` carrega e valida esse *bundle*, verificando se o modelo está treinado (por exemplo, se possui o atributo `classes_`). A função `predict_traffic` é responsável por selecionar do CSV apenas os fluxos ainda não processados, gerar um arquivo temporário com esses registros e repassar esse arquivo ao módulo `predict_knn`.

O módulo de predição aplica um pré-processamento adicional aos dados e converte identificadores textuais em representações numéricas por meio de *hashes* seguros, trata valores ausentes, substitui infinitos por valores válidos e normaliza as colunas numéricas usando o mesmo *scaler* do treinamento. Em seguida, o modelo k-NN é invocado para rotular cada fluxo como benigno ou malicioso. O resultado é retornado na forma de um vetor de predições, que também é acoplado a um *dataframe* para registro e análise.

As predições produzidas pelo k-NN são então agregadas pelo controlador. Cada fluxo recém-classificado é registrado na tabela `flows` do banco *SQLite*, incluindo o rótulo atribuído (legítimo ou malicioso) e o *hash* do fluxo, que permite rastrear alterações ao longo do tempo.

A mitigação é realizada na própria camada de controle SDN. Ao identificar um fluxo malicioso, o controlador chama o método `block_traffic`, que constrói uma regra de fluxo de alta prioridade (`priority = 65535`) e a envia via requisição HTTP ao *endpoint* `/stats/flowentry/add` do *Ryu*. Essa regra coincide com o par IP de origem e destino do fluxo suspeito e possui uma lista de ações vazia, o que faz com que os pacotes correspondentes sejam descartados. O bloqueio é persistido na tabela `blocked_flows`, com registro do *switch*, endereços envolvidos, horário e motivo. Para evitar repetição de comandos desnecessários, o sistema mantém em memória um dicionário de fontes já bloqueadas e respeita um intervalo mínimo entre tentativas de bloqueio para o mesmo par de IPs.

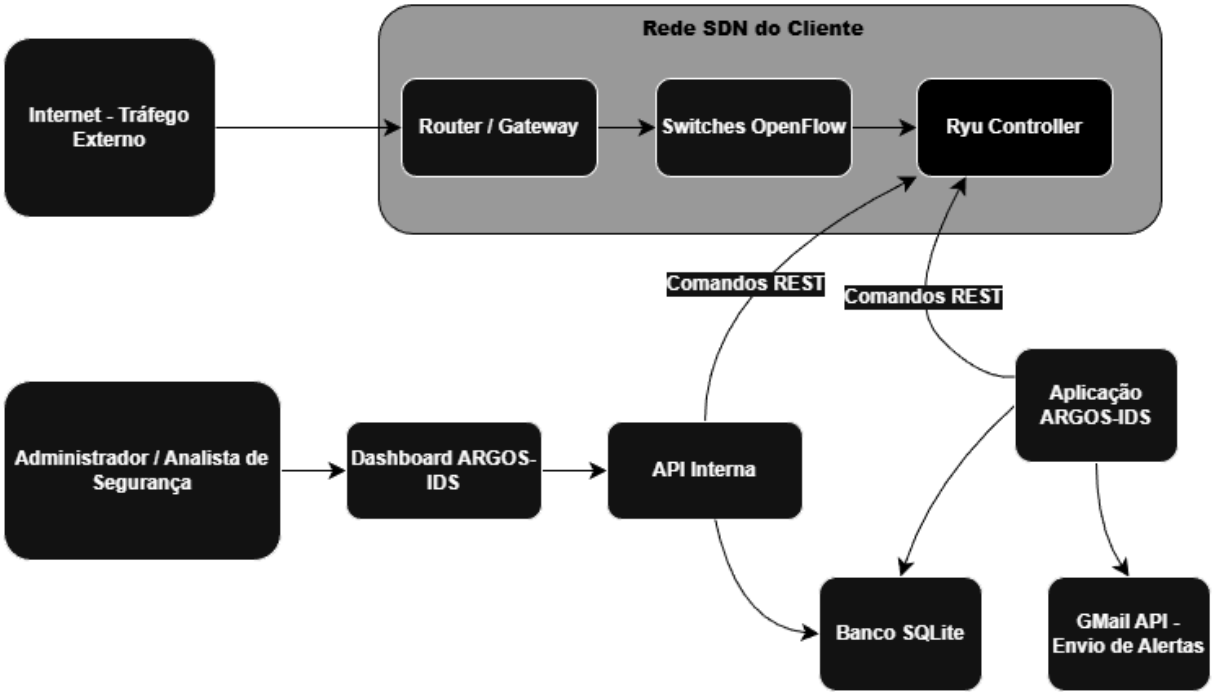
Além das ações na rede, o ARGOS-IDS pode acionar alertas assíncronos. A partir da

proporção de fluxos maliciosos observados em um determinado período, o sistema verifica se a taxa de tráfego malicioso supera um limite configurado. Quando isso ocorre, uma mensagem é gerada e enviada via *e-mail* aos contatos cadastrados, utilizando uma função auxiliar. A lista de destinatários ativos é obtida da tabela `alert_contacts`, permitindo ao administrador manter e gerenciar os canais de notificação sem alterar o código.

Por fim, o modo de mitigação adotado pelo controlador pode ser *block* ou *alert*. Em modo *block*, as regras de bloqueio são, de fato, instaladas nos *switches* SDN. Em modo *alert*, o sistema apenas registra e sinaliza os fluxos suspeitos, sem alterar o comportamento da rede. Essa configuração é consultada a cada ciclo de monitoramento e é definida pelo usuário por meio da *interface web*, conforme descrito na Seção 3.2.

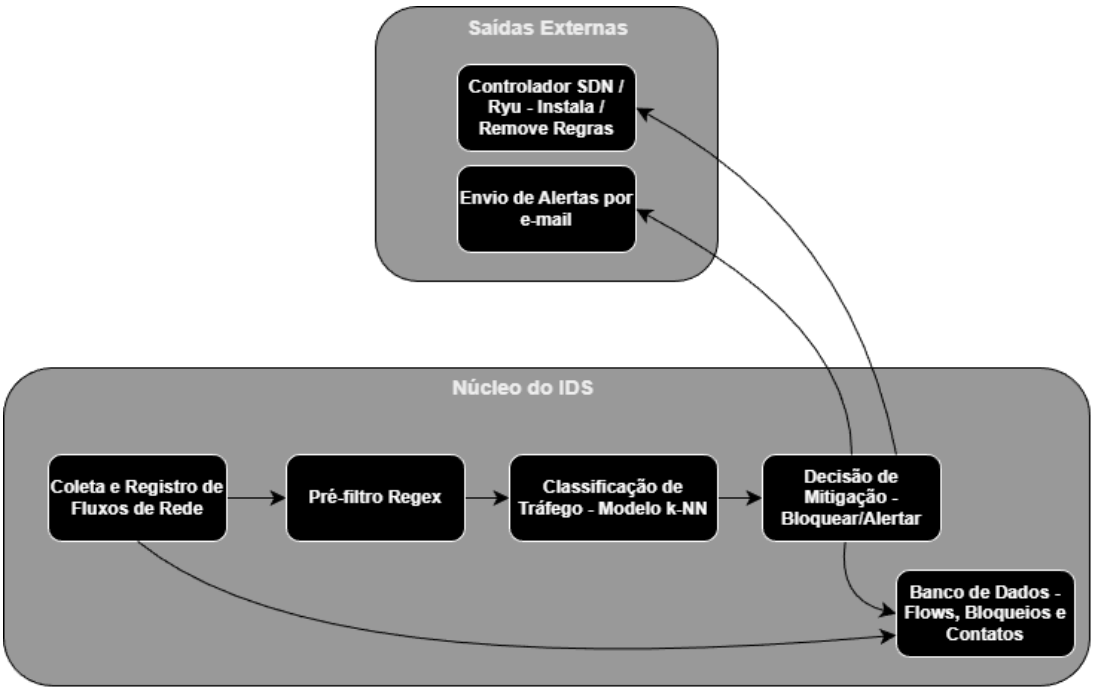
A Figura 2 apresenta uma visão geral do fluxo de dados, destacando a sequência da coleta até a entrega dos dados via interface, enquanto a Figura 3 amplia a visão do Argos-IDS e detalha a interação entre suas operações.

Figura 2 – Fluxo de dados do sistema



Fonte: Autora

Figura 3 – Aplicação Argos-IDS



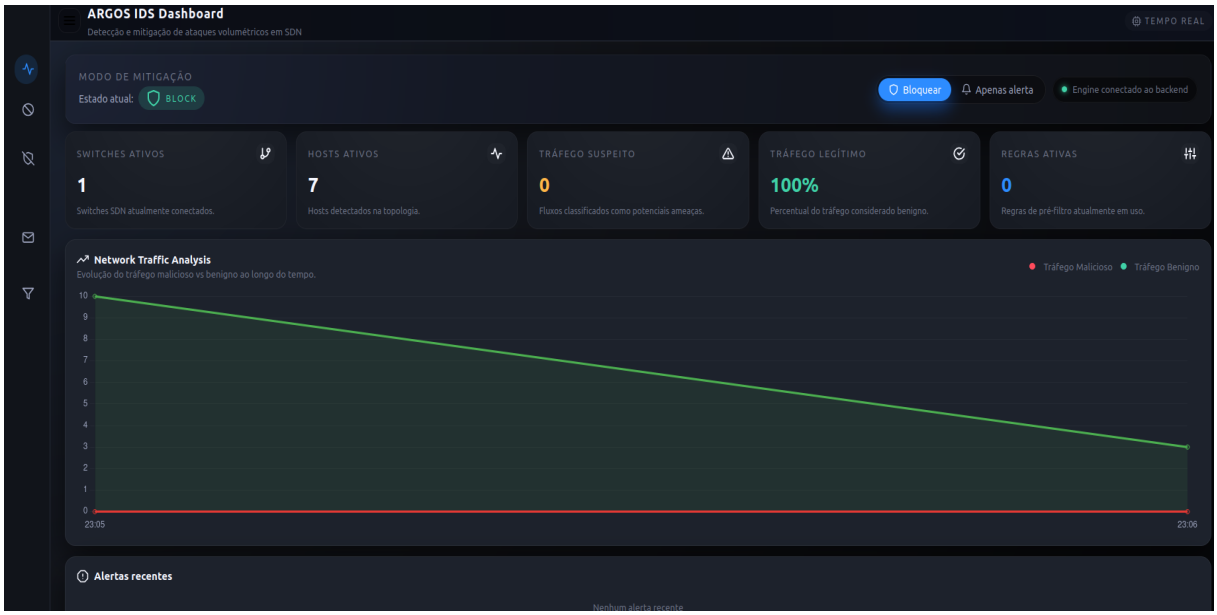
Fonte: Autora

3.2 Plataforma Interativa e Integração com o Ambiente do Usuário

Para tornar o uso do ARGOS-IDS acessível a administradores de rede com diferentes níveis de familiaridade em *cibersegurança*, foi desenvolvida uma plataforma interativa em formato de aplicação *web* local. Essa plataforma é servida por uma *API* implementada em *FastAPI*, que expõe um conjunto de *endpoints REST* responsáveis por consultar o banco de dados, interagir com o controlador *Ryu* e gerenciar configurações do sistema. O painel em si é construído em *HTML, CSS e JavaScript,* utilizando a biblioteca *Chart.js* para visualização gráfica.

Ao acessar o painel pelo navegador, o usuário visualiza um *dashboard* (visualizado na Figura 4)com cartões de resumo e gráficos de linha. Esses elementos são alimentados pelos endpoints `/api/traffic`, `/api/stats` e `/api/overview`, que, respectivamente, retornam a evolução temporal de fluxos benignos e maliciosos, estatísticas agregadas de detecção (como porcentagem de tráfego legítimo e número de bloqueios ativos) e uma visão geral da topologia SDN, incluindo o número de *switches* e *hosts* conectados. A lógica de agregação de dados para o gráfico organiza os fluxos em janelas de tempo, somando a quantidade de eventos benignos e maliciosos por intervalo, o que facilita a percepção de picos de ataque.

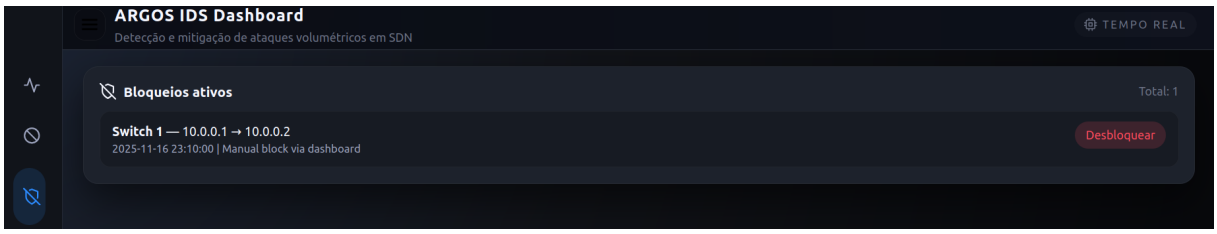
Figura 4 – Tela inicial Argos-IDS



Fonte: Autora

A plataforma também permite o controle direto da mitigação da rede. Por meio da interface de bloqueio, o usuário pode selecionar um *switch* e indicar um par de *IP* de origem e destino a ser bloqueado, acionando o *endpoint* `/block`, que, internamente, instala uma regra de bloqueio via *API* do *Ryu*. As entradas ativas de bloqueio são consultadas pelo *endpoint* `/api/blocked`, que lê a tabela `blocked_flows` e apresenta os registros no painel, com a possibilidade de desbloqueio individual na página de desbloqueio (Figura 5), o que dispara uma requisição de remoção de fluxo no controlador *SDN* e marca o bloqueio como inativo no banco.

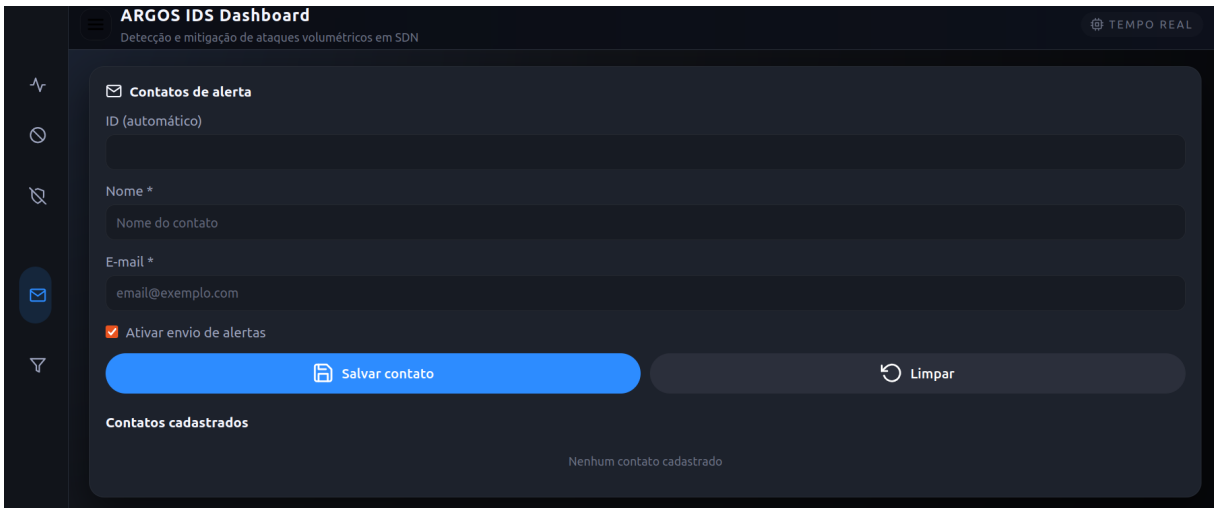
Figura 5 – Tela de Bloqueios



Fonte: Autora

Outro elemento central da plataforma é o gerenciamento de contatos de alerta. Os *endpoints* `/api/contacts` permitem listar, criar, atualizar e remover registros na tabela `alert_contacts`. O painel oferece um formulário simples para cadastro de nome, *e-mail* e ativação do contato, fornecendo ao controlador uma lista de destinatários sempre atualizada para o envio de notificações por *e-mail* quando são detectados níveis anormais de tráfego malicioso ou quando novas regras de bloqueio são aplicadas (Figura 6).

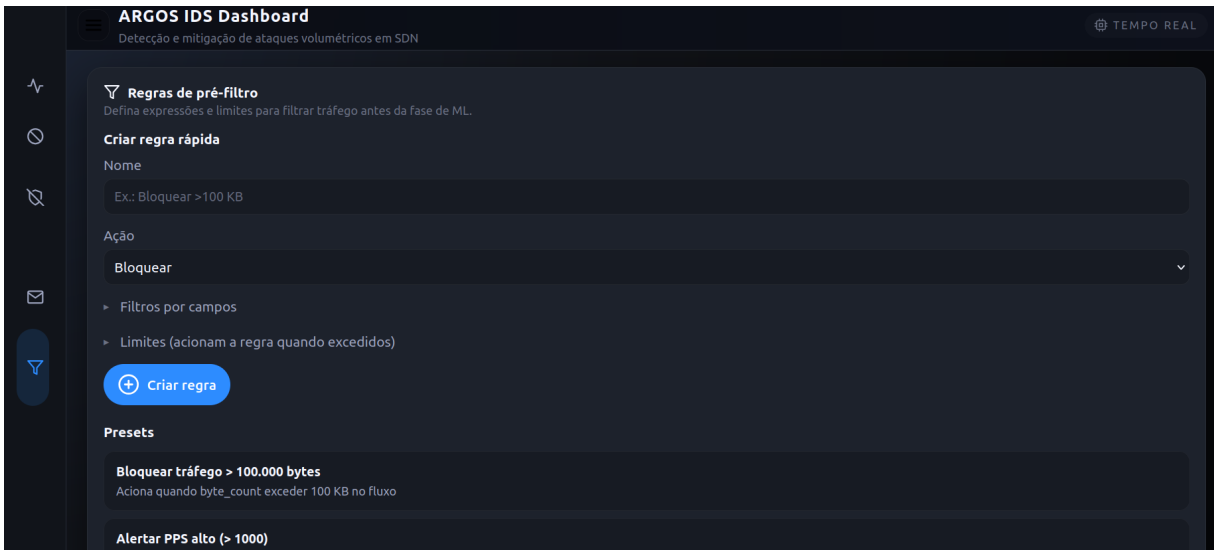
Figura 6 – Tela de Configuração de Contatos



Fonte: Autora

A configuração de regras de pré-filtro também é feita via painel (Figura 7). A *API* expõe *endpoints* para criação, listagem, ativação, desativação e exclusão de regras na tabela `filter_rules`. Cada regra pode ser definida com nome, descrição, padrões de *IP* (que serão convertidos em expressões regulares no controlador), protocolo, portas de origem e destino e limites volumétricos de *bytes*, pacotes, *PPS* e *BPS*, além da ação desejada (*block* ou *alert*). A interface inclui ainda um conjunto de *presets* que servem como base para criação rápida de regras típicas, como bloqueio de portas inseguras ou detecção de tráfego *ICMP* de *broadcast*. Dessa forma, o administrador pode ajustar o comportamento do IDS sem necessidade de editar arquivos de configuração ou reiniciar o serviço.

Figura 7 – Tela de Configuração de Pré-Filtros



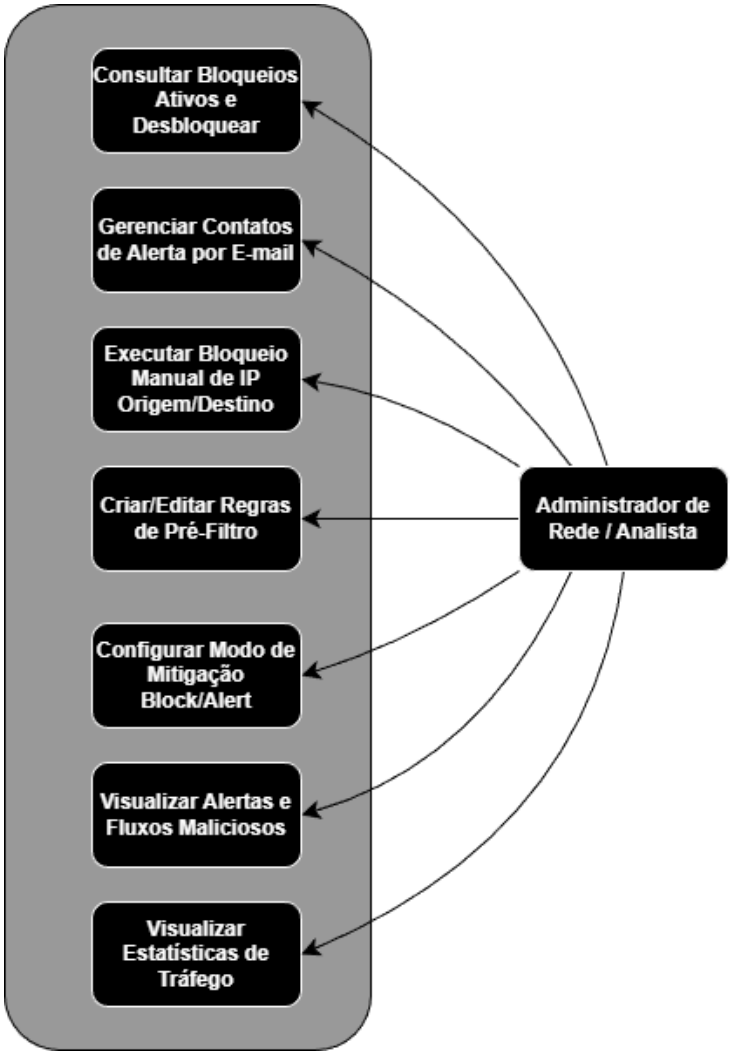
Fonte: Autora

O modo de mitigação global (bloqueio ou alerta apenas) também é configurado pela

plataforma. A aplicação mantém um arquivo de configuração local (`mitigation_mode.json`), acessado pelos *endpoints* `/api/config/mode` *(GET e POST)*. O painel oferece botões para alternar entre *BLOCK* e *ALERT*, e o controlador consulta esse valor em cada ciclo de monitoramento por meio de uma requisição *HTTP* à *API* do painel. Quando o modo *alert* está ativo, os eventos suspeitos são registrados e exibidos, mas as regras de bloqueio não são instaladas na rede, o que é útil em fases de calibração ou validação do sistema.

Uma visão geral da interface e suas funcionalidades são apresentadas na Figura 8.

Figura 8 – Diagrama de Casos de Uso



Fonte: Autora

Essa metodologia de implementação, ancorada na divisão clara de responsabilidades entre coleta, pré-filtro, classificação, mitigação, persistência e interface, garante que o ARGOS-IDS possa ser evoluído ao longo do tempo, com a substituição de modelos de aprendizado de máquina, ajuste de heurísticas ou inclusão de novos tipos de visualização, sem a necessidade de reestruturar todo o sistema.

4 Tecnologias Utilizadas

O desenvolvimento do sistema ARGOS-IDS foi estruturado com base em um conjunto abrangente de tecnologias modernas, escolhidas estrategicamente para garantir desempenho, portabilidade, facilidade de integração e aderência ao contexto de redes definidas por *software* (SDN). Este capítulo descreve as linguagens, bibliotecas, *frameworks* e ferramentas utilizadas em cada um dos módulos do sistema, bem como a infraestrutura que permite sua operação.

4.1 Linguagens de Programação

A escolha das linguagens de programação no ARGOS-IDS teve como critério principal o desenvolvimento de redes compatível com o controlador *RYU*, compatibilidade com bibliotecas e facilidade de manutenção:

- **Python:** é a linguagem principal que será utilizada em todo o backend do sistema. Ela vai ser empregada tanto no desenvolvimento do IDS quanto no controlador SDN baseado no *framework Ryu*. Sua sintaxe simples e a grande variedade de bibliotecas para ciência de dados, redes e *APIs REST* tornam Python uma escolha ideal (PYTHON, 2025).
- **HTML, CSS e JavaScript:** utilizadas no desenvolvimento da interface *web* interativa. Essas tecnologias permitem a criação de páginas dinâmicas e responsivas que exibem informações em tempo real sobre o tráfego de rede e os alertas de segurança.

4.2 *Frameworks* e Bibliotecas

- **Ryu:** um *framework* de controlador SDN escrito em Python que permite o desenvolvimento rápido de aplicativos de controle. No ARGOS-IDS, o Ryu é responsável por aplicar regras de bloqueio ou redirecionamento com base nas decisões tomadas pelo IDS, além de fornecer informações sobre o estado atual da rede (RYU. . . , 2025).
- **Scikit-learn:** biblioteca de aprendizado de máquina usada no desenvolvimento e aplicação dos classificadores de tráfego. Os modelos como *Naïve Bayes, Random Forest*, Árvore de Decisão, SVM e k-NN foram todos treinados com essa biblioteca e são carregados para inferência em tempo real (SCIKIT-LEARN, 2025).
- **FastAPI:** *microframework web* utilizado para criar a *API* local que faz a ponte entre o IDS e a interface *web*. Essa API permite que a plataforma interativa envie comandos para o IDS e receba alertas e estatísticas (FAST. . . , 2025).

- **Regex (re):** módulo padrão do Python para o uso de expressões regulares, essencial para o pré-filtro de pacotes no controlador SDN, descartando tráfego malicioso de forma eficiente antes da classificação por *machine learning* (REGEX. . . , 2025).
- **Chart.js:** biblioteca JavaScript para geração de gráficos interativos na interface *web*. É usada para exibir visualizações em tempo real do volume de tráfego legítimo e malicioso (CHART. . . , 2025).

4.3 Banco de Dados

- **SQLite:** um sistema de banco de dados leve, relacional e embutido. No ARGOS-IDS, ele armazena decisões de mitigação, dados de *login* de usuários e configurações do sistema. Por ser local e não exigir a instalação de um servidor separado, o SQLite é ideal para ambientes que priorizam portabilidade e simplicidade (SQLITE, 2025).

4.4 APIs

A comunicação e coleta de dados do Argos-IDS é baseada em *APIs REST,* possibilitando uma arquitetura modular e distribuída. Essa abordagem permite que os diferentes componentes do sistema operem de forma integrada e flexível, mantendo a escalabilidade e a facilidade de manutenção.

API do Controlador SDN (Ryu)

O ARGOS-IDS utiliza o módulo `ofctl_rest` do controlador Ryu, que disponibiliza uma API REST para ações de controle de rede. Essa API permite a instalação, modificação e remoção de regras de fluxo em *switches OpenFlow*. O módulo de decisão do IDS se comunica com essa API para aplicar medidas de mitigação, como bloqueio ou redirecionamento de pacotes (RYU. . . , 2025).

API do Google *Mail* (SMTP)

Para o envio automático de alertas por *e-mail*, o sistema utiliza a API SMTP do Gmail. A plataforma interativa, por meio do Módulo de Envio de *E-mails*, se conecta ao servidor SMTP do Google autenticando-se com as credenciais configuradas previamente. Ao detectar um ataque, o sistema gera uma mensagem e envia notificações diretamente aos usuários responsáveis pela rede. O envio é feito de forma automatizada e segura, respeitando os protocolos TLS/SSL (GOOGLE. . . , 2025).

4.5 Ambiente de Testes

A combinação das ferramentas *Ryu* e *Mininet* é fundamental para o desenvolvimento e validação do sistema ARGOS-IDS, permitindo a criação e o teste de redes definidas por software (SDN) de forma eficiente e em um ambiente controlado.

4.5.1 Ryu

Ryu é um *framework* de controlador SDN escrito em Python que facilita o desenvolvimento de aplicações de controle para redes baseadas no protocolo *OpenFlow*. Ele permite que os desenvolvedores implementem facilmente funcionalidades de controle, como roteamento, balanceamento de carga e monitoramento de tráfego (RYU-CONTROLLER, 2025). No contexto do ARGOS-IDS, o Ryu desempenha um papel crucial ao interagir com a infraestrutura de rede, aplicando regras de mitigação com base nas decisões do IDS.

4.5.2 Mininet

O *Mininet* é uma ferramenta poderosa para a emulação de redes virtuais, essencial para testar e validar redes SDN de forma eficiente. Ele cria redes de *hosts*, *switches* e controladores virtuais, permitindo que os desenvolvedores simulem e experimentem com diferentes topologias de rede e comportamentos de tráfego sem a necessidade de uma infraestrutura física complexa (MININET, 2025). No ARGOS-IDS, o Mininet é utilizado para simular o tráfego de rede e verificar o desempenho do IDS em um ambiente controlado antes de sua implementação em um ambiente real.

5 Validação e Testes

Esta seção descreve a validação prática do ARGOS-IDS em um ambiente SDN emulado, com foco no comportamento do sistema durante a detecção e mitigação de ataques, no funcionamento da interface web e no envio de alertas por e-mail. Mais do que medir apenas métricas de acurácia, o objetivo aqui foi observar o comportamento integrado do sistema em cenários realistas de tráfego benigno e malicioso, registrando o efeito das decisões do IDS sobre o controlador SDN e sobre a visualização no painel interativo.

Os testes foram realizados utilizando o controlador Ryu, em conjunto com um ambiente de rede emulado no Mininet. O controlador foi configurado com o aplicativo *ofctl_rest*, permitindo a consulta a estatísticas de fluxo e a instalação de regras de bloqueio via API REST. Em paralelo, a API web desenvolvida em *FastAPI* e a interface HTML/CSS/JavaScript do ARGOS-IDS foram executadas localmente, consumindo o banco de dados *SQLite* gerado pelo controlador.

5.1 Cenário 1: Tráfego Benigno e Operação Estável do Sistema

No primeiro cenário, o ambiente SDN foi inicializado com um conjunto reduzido de hosts emulando uma rede simples. O controlador Ryu, iniciou o processo de monitoramento periódico, realizando requisições à URL `/stats/flow` a cada ciclo de execução. Os fluxos retornados pela API REST foram convertidos em um formato tabular, armazenados em arquivo *CSV* e, simultaneamente, gravados na tabela `flows` do banco *traffic.db*.

Para este cenário, foi gerado apenas tráfego benigno (como *pings* e comunicações simples entre hosts). O modelo k-NN previamente treinado foi aplicado sobre os fluxos recém-coletados, produzindo uma classificação binária (benigno ou malicioso) para cada linha do arquivo de predição. Os registros foram salvos com o rótulo correspondente no banco de dados e nenhuma regra de bloqueio foi instalada, uma vez que o tráfego foi classificado predominantemente como legítimo (Figura 9).

Figura 9 – Simulação de Tráfego Legítimo

Fonte: Autora

No painel web, o dashboard exibiu, em tempo real, a quantidade de switches e hosts ativos detectados via API do Ryu, bem como o percentual de tráfego legítimo. A curva apresentada no gráfico de tráfego mostrou apenas pequenas variações benignas ao longo do tempo, sem registros de bloqueios ou alertas. Esse cenário serve como referência de operação estável do sistema, com o IDS ativo, mas sem interferir no tráfego por ausência de ataques.

5.2 Cenário 2: Detecção e Bloqueio Automático de Ataque Volumétrico

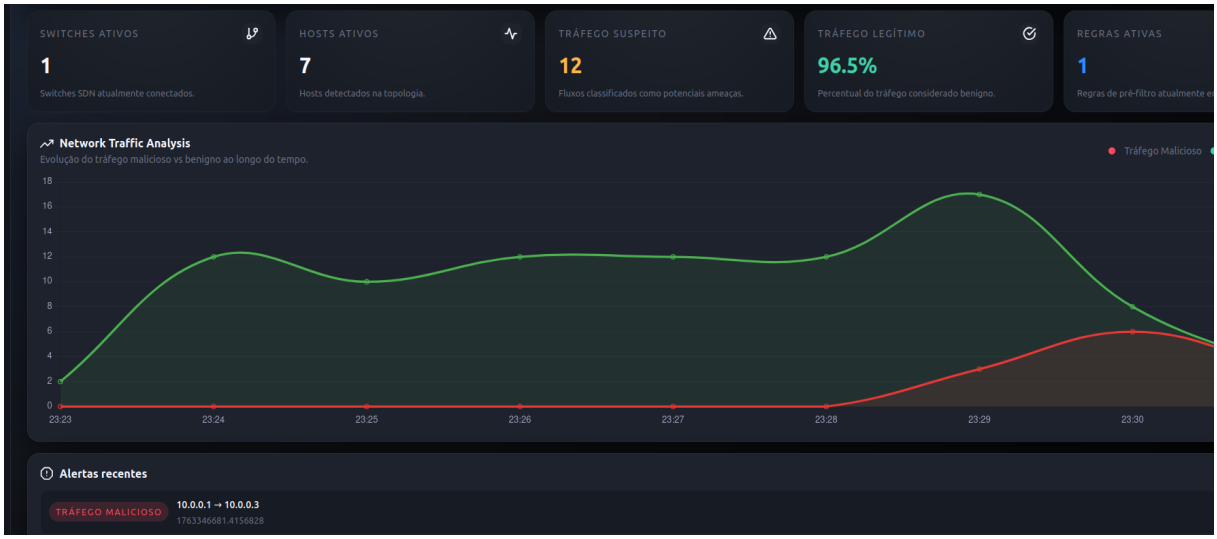
No segundo cenário, o foco passou a ser a capacidade do ARGOS-IDS de detectar No segundo cenário, o objetivo foi validar a capacidade do ARGOS-IDS de detectar e bloquear automaticamente um ataque de alta taxa de transferência. Para simular um ataque volumétrico, foi executado um fluxo UDP de alta intensidade no Mininet:

```
h1 iperf -c 10.0.0.3 -u -b 1000M -t 20
```

O uso do modo *UDP* com largura de banda de 1 Gbps faz com que o host envie pacotes na maior velocidade possível, reproduzindo o comportamento de um ataque de negação de serviço por inundação.

Durante a simulação, o controlador coletou fluxos com taxas muito acima das normais. Em seguida, o pipeline de classificação confirmou o comportamento malicioso e o módulo de mitigação instalou uma regra de bloqueio diretamente no switch OpenFlow (Figura 10).

Figura 10 – Simulação de Tráfego Maligno



Fonte: Autora

O efeito desse bloqueio pôde ser observado em três camadas: na rede emulada, onde o host atacante deixou de conseguir alcançar o destino; nos logs do controlador, que passaram a registrar a instalação da regra e a detecção do ataque; e no dashboard web, que passou a exibir a lista de bloqueios ativos e um aumento súbito no volume de tráfego malicioso seguido de estabilização após a mitigação. Na Figura 11 é simulado tráfego entre os hosts, e é evidenciado o bloqueio entre os hosts após o ataque.

Figura 11 – Simulação de Tráfego entre os Hosts

```
*** Ping: testing ping reachability
h1 -> h2 X h4
h2 -> h1 h3 h4
h3 -> X h2 h4
h4 -> h1 h2 h3
*** Results: 16% dropped (10/12 received)
```

Fonte: Autora

5.3 Cenário 3: Modo “Apenas Alerta” sem Bloqueio de Tráfego

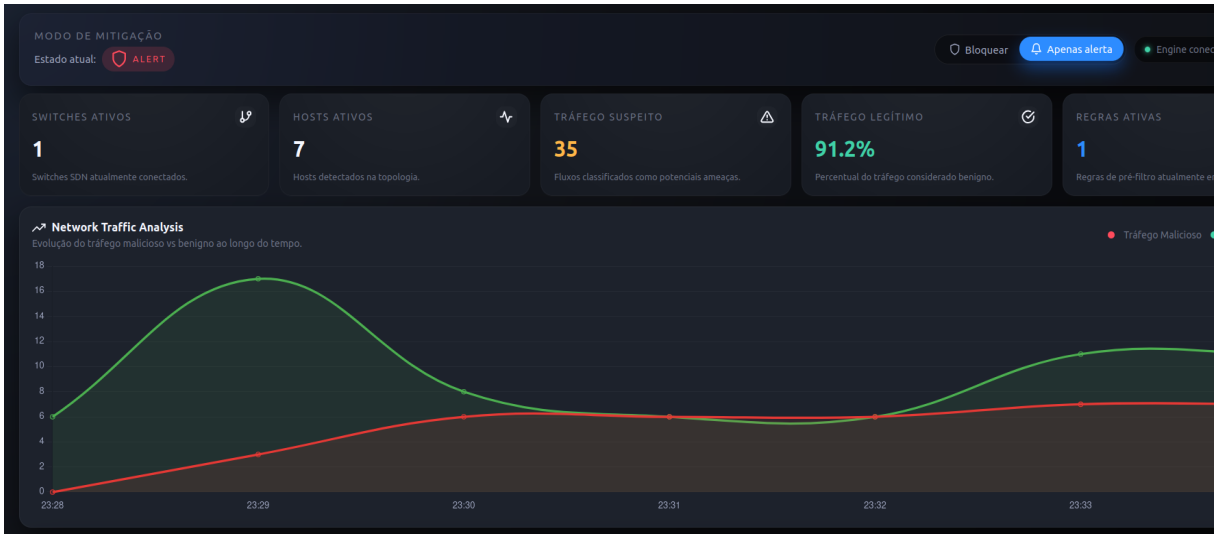
Um aspecto importante da validação envolveu o teste do modo de operação em que o ARGOS-IDS não aplica bloqueios automaticamente, mas apenas registra e sinaliza os eventos suspeitos. Esse comportamento é configurado pelo *dashboard*, que expõe um controle de “modo de mitigação” por meio de uma chamada à API `/api/config/mode`. Ao selecionar o modo *alert*,

o arquivo de configuração local é atualizado e o controlador passa a ler esse valor periodicamente no método de monitoramento.

Com o sistema nesse modo, um novo ataque volumétrico foi disparado no Mininet. O pipeline de detecção permaneceu o mesmo: coleta de fluxos, cálculo de métricas, classificação com k-NN e verificação de limites heurísticos. A diferença está na etapa de mitigação: ao chegar no método `block_traffic`, o controlador verifica o modo ativo e, se estiver em *alert only*, o sistema registra o ataque, gera logs e pode acionar contatos por e-mail, mas não instala regras de bloqueio nos switches.

Na interface web, esse cenário é mostrado por meio de um gráfico com aumento de tráfego malicioso, combinado com alertas destacados na seção de notificações recentes, porém sem incremento na contagem de bloqueios ativos (Figura 12). Esse comportamento é útil em ambientes onde o administrador ainda está observando o comportamento do IDS e prefere não automatizar o bloqueio, utilizando o sistema inicialmente apenas como ferramenta de monitoração.

Figura 12 – Argos-IDS em modo Alerta



Fonte: Autora

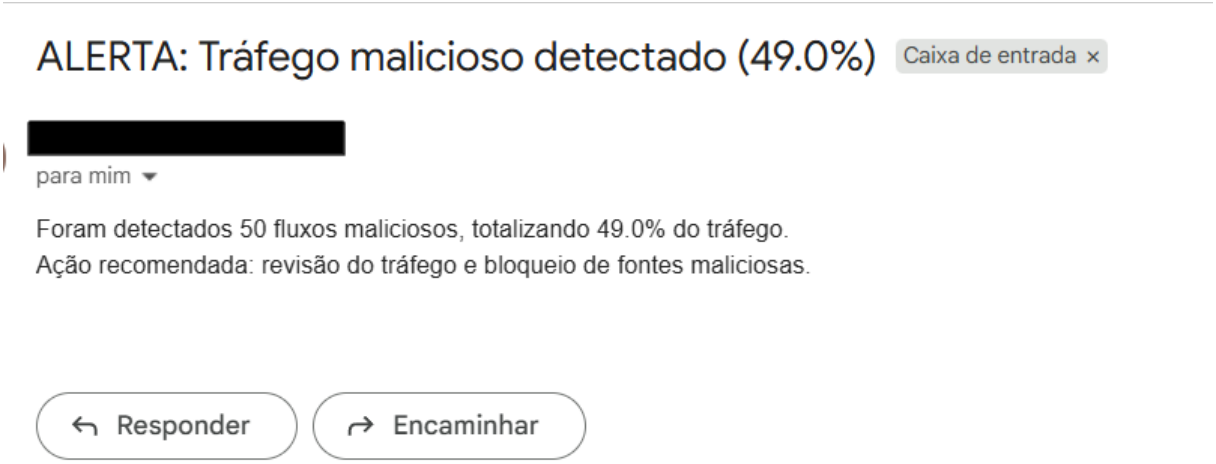
5.4 Cenário 4: Envio de Alertas por E-mail

Outro aspecto observado na validação foi o envio automático de e-mails de alerta para contatos previamente configurados. Os endereços de e-mail são armazenados na tabela `alert_contacts`, gerenciada pelo painel web por meio dos endpoints `/api/contacts`. Na fase de detecção, o controlador consulta os contatos habilitados e, quando determinada condição de risco é atingida, aciona a função `send_gmail` para disparar mensagens (Figura 13).

Na prática, foram configurados um ou mais contatos de teste e repetido o cenário de ataque volumétrico. Quando a taxa de tráfego malicioso ultrapassou um limiar pré-definido ou quando uma regra específica foi acionada, o sistema gerou um e-mail contendo um resumo da

ocorrência, incluindo IP de origem, IP de destino, dpID do switch e horário aproximado do evento. Os e-mails chegaram à caixa de entrada dos destinatários, confirmando a integração entre o controlador, a API web e o serviço de envio de mensagens.

Figura 13 – Email enviado pelo Argos-IDS



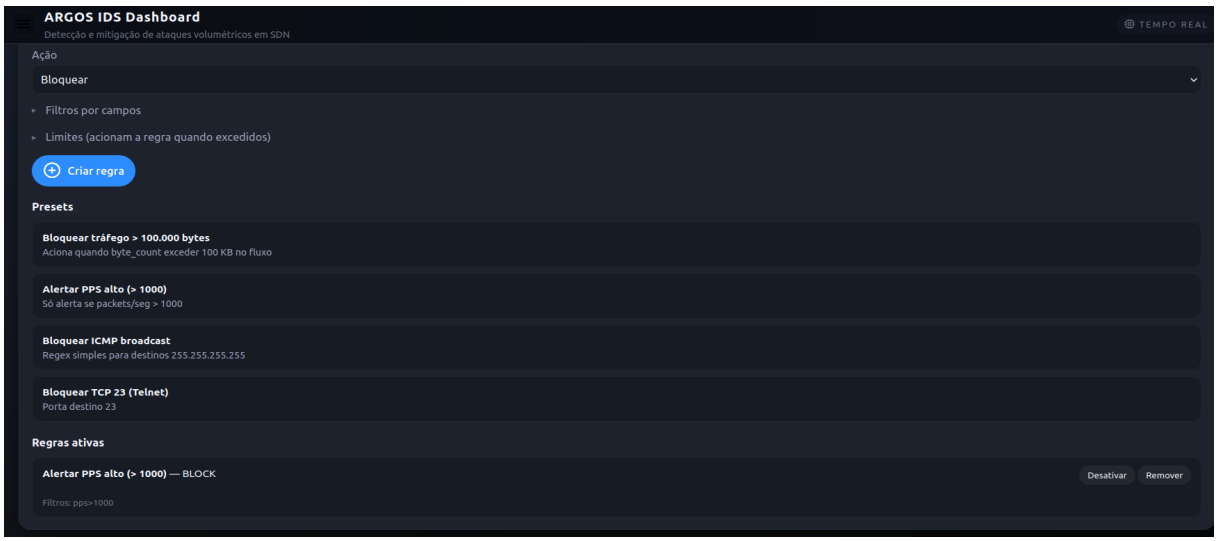
Fonte: Autora

5.5 Cenário 5: Uso de Regras de Pré-Filtro na Prática

Por fim, foi validada a utilização de regras de pré-filtro definidas pelo usuário para reduzir o volume de tráfego encaminhado ao modelo de classificação. Essas regras são cadastradas na interface web, que envia os dados para a API em `/api/rules`, armazenando-os na tabela `filter_rules`. Cada regra pode especificar endereços IP de origem e destino (com suporte a padrões em formato de expressão regular), protocolo, portas de origem e destino, bem como limites mínimos de bytes, pacotes, PPS e BPS.

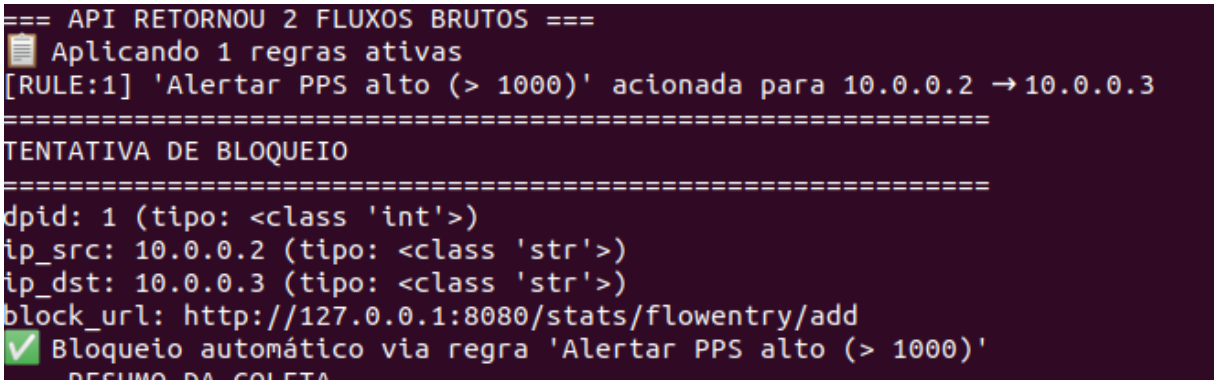
Durante os testes, foram criadas regras simples, como bloqueio de tráfego TCP em determinada porta ou alerta para fluxos com volume superior a um valor específico (Figura 14). Em seguida, tráfego artificial correspondente a esses padrões foi gerado no Mininet. O controlador, ao coletar os fluxos, aplicou as regras compiladas na função `_flow_matches_rule`, que verifica progressivamente se o fluxo atende aos critérios definidos. Quando uma regra com ação *block* estava ativa e o modo de mitigação configurado para *block*, o sistema instalou imediatamente a regra de bloqueio no switch, antes mesmo da etapa de classificação por k-NN (Figura 15). Quando a ação da regra era *alert*, o sistema apenas gerava registro e permitia o fluxo seguir.

Figura 14 – Adição de Regras de Pré-Filtro



Fonte: Autora

Figura 15 – Pré-Filtro Bloqueando o Tráfego



Fonte: Autora

Esse mecanismo de pré-filtro mostrou-se especialmente útil para reduzir o trabalho do modelo em situações de ataques volumétricos massivos, além de permitir que o administrador da rede traduza políticas de segurança simples em regras diretas no painel. Um print da tela de gerenciamento de regras, ao lado do painel de bloqueios ativos, ilustra como esses elementos funcionam em conjunto na operação cotidiana do ARGOS-IDS.

5.6 Síntese dos Testes

Em conjunto, os cenários experimentais demonstram que o ARGOS-IDS é capaz de monitorar o tráfego em tempo real, classificar fluxos com base em um modelo de aprendizado de máquina treinado previamente, aplicar heurísticas de detecção volumétrica, instalar regras de bloqueio no plano de dados, operar em modo apenas de alerta, enviar notificações por e-mail e permitir a definição de regras de pré-filtro pelo usuário. A utilização combinada dos logs

do controlador, dos registros no banco de dados e das telas do dashboard fornece uma visão consistente do comportamento do sistema.

6 Conclusões

O desenvolvimento do ARGOS-IDS demonstrou que é possível construir uma solução de detecção e mitigação de ataques volumétricos voltada a ambientes SDN com baixo custo, alta compatibilidade e operação simplificada. O sistema foi criado para atender principalmente redes pequenas e médias que não dispõem de equipe ou ferramentas especializadas em *cibersegurança*, e os resultados obtidos ao longo da implementação e validação confirmam que esse objetivo foi alcançado.

Durante os testes realizados em ambiente emulado, utilizando tráfego gerado com *iperf3*, o ARGOS-IDS se mostrou capaz de monitorar fluxos de forma contínua, identificar padrões anômalos e classificar ataques com precisão por meio do modelo k-NN treinado. Quando configurado para mitigação ativa, o sistema aplicou bloqueios diretamente nos *switches OpenFlow*, interrompendo imediatamente os fluxos maliciosos. Nos cenários em que foi utilizado apenas em modo de alerta, o sistema registrou os eventos, atualizou o *dashboard* e enviou notificações por *e-mail* para os contatos cadastrados. Todas essas etapas foram refletidas em tempo real na interface *web*, evidenciando a integração sólida entre os módulos de *backend*, o controlador SDN e o painel de administração.

Ao longo do desenvolvimento, algumas decisões precisaram ser ajustadas em razão do tempo disponível para o projeto. A principal mudança foi a opção por utilizar, nesta versão, apenas o modelo k-NN durante a fase de inferência, ainda que o sistema tenha sido originalmente concebido para trabalhar com múltiplos classificadores e realizar uma votação ponderada entre eles. Embora esse mecanismo já esteja parcialmente implementado na estrutura interna do código, sua integração completa foi planejada como etapa futura. Outra adaptação importante que não foi implementada, foi a inclusão de um mecanismo de autenticação simples na interface *web*, que se mostra necessária para impedir modificações indevidas em configurações sensíveis do sistema.

Além dos resultados técnicos, o desenvolvimento do ARGOS-IDS proporcionou um aprendizado significativo em relação ao funcionamento de redes definidas por software, à comunicação entre *switches* e controlador e à lógica de instalação de regras de fluxo em *OpenFlow*. A experiência prática com o *pipeline* completo de captura, análise e tomada de decisão permitiu compreender de forma mais profunda como sistemas de detecção de intrusão operam em nível de rede e como podem ser integrados a ambientes SDN reais.

Embora o sistema já apresente um conjunto sólido de funcionalidades, há um caminho natural para sua evolução. Entre as principais direções estão a integração plena de múltiplos modelos de aprendizado de máquina com votação ponderada, o aprimoramento do mecanismo de autenticação e permissão, a inclusão de métricas avançadas no *dashboard* e o aprofundamento da análise temporal dos ataques. Melhorias também podem envolver técnicas mais sofisticadas

de correlação de eventos, permitindo identificar ataques distribuídos com múltiplas origens simultâneas.

Por fim, destaca-se que o ARGOS-IDS não apenas funciona como um projeto acadêmico, mas também apresenta potencial real de empacotamento e distribuição como solução prática. Sua arquitetura modular, a dependência reduzida de infraestrutura e a interface *web* intuitiva permitem que ele seja facilmente instalado, configurado e utilizado por clientes finais, especialmente em pequenas empresas que necessitam fortalecer sua segurança sem elevar custos operacionais. Com pequenas adaptações e documentação orientada ao usuário, o sistema pode evoluir para um produto utilizável em ambientes de produção.

Diante dos resultados e da experiência adquirida, conclui-se que o ARGOS-IDS cumpre sua proposta inicial ao oferecer uma alternativa viável, acessível e eficaz para aumentar a resiliência de redes SDN contra ataques volumétricos. O projeto também evidencia o potencial de crescimento da solução, reforçando sua contribuição para a democratização da cibersegurança e para a formação técnica associada ao domínio de redes e sistemas de detecção de intrusão.

Referências

ALAWADI, A. *SNT (Simulated Network Traffic Using Mininet and Ryu) DDoS Detection Dataset*. 2025. Acesso em: 29 outubro 2025. Disponível em: <https://data.mendeley.com/datasets/9hz6f62gtk/1>.

CHART.JS. 2025. Acesso em: 30 junho 2025. Disponível em: <https://www.chartjs.org/>.

CHOU, E.; GROVES, R. *Distributed Denial of Service (DDoS)*. [S.l.]: O'Reilly Media, Inc, 2018.

DISTRIBUTED denial-of-service (DDoS) protection. 2025. Acesso em: 01 abril 2025. Disponível em: <https://www.cloudflare.com/ddos/>.

FAST API. 2025. Acesso em: 30 junho 2025. Disponível em: <https://fastapi.tiangolo.com/>.

GOOGLE Mail API. 2025. Acesso em: 30 junho 2025. Disponível em: <https://developers.google.com/workspace/gmail/api/guides?hl=pt-br>.

MININET. 2025. Acesso em: 29 novembro 2025. Disponível em: <https://mininet.org/>.

NADEEM HOCK GUANGOH, V. P. Y. A. M. W. Ddos detection in sdn using machine learning techniques. 2021. Acesso em: 29 março 2025. Disponível em: <https://www.researchgate.net/publication/357269421_DDoS_Detection_in_SDN_using_Machine_Learning_Techniques>.

NERY, C. Internet foi acessada em 72,5 milhões de domicílios do país em 2023. 2023. Acesso em: 29 março 2025. Disponível em: <https://agenciadenoticias.ibge.gov.br/agencia-noticias/2012-agencia-de-noticias/noticias/41024-internet-foi-acessada-em-72-5-milhoes-de-domicilios-do-pais-em-2023>.

NIN, C. S. Ddos trends: Volumetric attacks are on the rise. *RCR Wireless News*, 2023. Acesso em: 29 março 2025. Disponível em: <https://www.rcrwireless.com/20230214/security/ddos-trends-volumetric-attacks-are-on-the-rise>.

PYTHON. 2025. Acesso em: 30 junho 2025. Disponível em: <https://www.python.org/>.

REGEX Python. 2025. Acesso em: 30 junho 2025. Disponível em: <https://docs.python.org/3/library/re.html>.

RYU-CONTROLLER. 2025. Acesso em: 29 novembro 2025. Disponível em: <https://ryu.readthedocs.io/en/latest/getting_started.html>.

RYU manager. 2025. Acesso em: 30 junho 2025. Disponível em: <https://ryu.readthedocs.io/en/latest/index.html>.

SCIKIT-LEARN. 2025. Acesso em: 30 junho 2025. Disponível em: <https://scikit-learn.org/stable/>.

SMALL and medium business (SMB) cyberattacks are frequent and costly. 2024. Acesso em: 28 outubro 2025. Disponível em: <https://cdn-dynmedia-1.microsoft.com/is/content/microsoftcorp/microsoft/final/en-us/microsoft-brand/documents/SMBCybersecurity-Report-Final.pdf>.

SQLITE. 2025. Acesso em: 30 junho 2025. Disponível em: <https://sqlite.org/>.

STAMUS Networks. 2025. Acesso em: 01 abril 2025. Disponível em: <https://www.stamus-networks.com/>.

UIT: uso da internet aumenta, mas disparidades permanecem. 2023. Acesso em: 29 março 2025. Disponível em: <https://www.abranet.org.br/publicacoes/noticias/5350>.