UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Eduardo Rafael Hirt

DESENVOLVIMENTO DE UMA INTERFACE WEB PARA MONITORAMENTO DE TRÁFEGO DE REDE

Santa Maria, RS
2019

Eduardo Rafael Hirt

DESENVOLVIMENTO DE UMA INTERFACE WEB PARA MONITORAMENTO DE TRÁFEGO DE REDE

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação, Área de Concentração em Software Básico, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Bacharel em Ciência da Computação**.

ORIENTADOR: Prof. João Vicente Ferreira Lima

463
Santa Maria, RS
2019

End. Eletr.: eduardohirt3@gmail.com

Eduardo Rafael Hirt

DESENVOLVIMENTO DE UMA INTERFACE WEB PARA MONITORAMENTO DE TRÁFEGO DE REDE

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação, Área de Concentração em Software Básico, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Bacharel em Ciência da Computação**.

Aprovado em 4 de dezembro de 2019:

João Vicente Ferreira Lima, Dr. (AAAA)
(Presidente/Orientador)

Benhur de Oliveira Stein, Dr. (UFSM)

Andrea Schwertner Charão, Dra. (UFSM)

Santa Maria, RS
2019

RESUMO

DESENVOLVIMENTO DE UMA INTERFACE WEB PARA MONITORAMENTO DE TRÁFEGO DE REDE

AUTOR: Eduardo Rafael Hirt
ORIENTADOR: João Vicente Ferreira Lima

Este trabalho apresenta o desenvolvimento de uma interface web para monitoramento de tráfego de rede. Esta visa atender a necessidade de visualização dos dados de sistema já existente, que preenche uma base de dados com informações relacionadas ao comportamento e o estado de uma rede. Para isto, foi desenvolvida uma API, que faz buscas à base de dados já existente, e, ligada a ela, uma interface web. Esta interface faz requisições à API e, utilizando conceitos de visualização de dados, exibe as informações retornadas de maneira gráfica. Ela possui gráficos interativos, suporta filtro dos dados por tempo e permite salvar uma imagem dos gráficos gerados.

Palavras-chave: Sistema Web. Monitoramento de Rede. Visualização de dados.

ABSTRACT

DEVELOPMENT OF A WEB INTERFACE FOR NETWORK TRAFFIC MONITORING

AUTHOR: Eduardo Rafael Hirt
ADVISOR: João Vicente Ferreira Lima

This work presents the development of a web interface for network traffic monitoring. It aims to meet the need of data visualization of the data obtained by an already existing system, that fills a database with information regarding the behavior and state of a network. For that, was developed an API, that searches the given database, and, connected to it, a web interface. That interface makes requests to the API and, utilizing concepts of data visualization, shows the returned information graphically. It has interactive graphics, supports data filter by time and allows the user to save an image of the generated graphics.

Keywords: Web system. Network monitoring. Data visualization.

LISTA DE FIGURAS

SUMÁRIO

1 INTRODUÇÃO

A relevância da internet e a dependência global relacionada a ela estão crescendo constantemente, o que resulta em consequências substanciais no caso de falhas de serviço, ataques bem sucedidos, dados roubados ou serviços interrompidos. Juntamente, uma tendência a ataques de rede massivos é perceptível globalmente, principalmente as que são feitas por dispositivos que realizam ataques automatizados de larga escala (GOODALL; CONTI; MA, 2008).

Em redes de grande escala, como a da Universidade Federal de Santa Maria (UFSM), que possui mais de 30 mil dependentes, incluindo alunos, docentes e servidores Técnicos Administrativos em Educação (CPD-UFSM, 2019), a prevenção de instabilidades e ataques se torna essencial. Impossibilidade de acesso a serviços, dados confidenciais expostos e dados oficiais alterados são apenas alguns dos possíveis resultados de uma intrusão bem sucedida.

O monitoramento de redes se torna essencial neste cenário. Ele traz informações sobre o estado da rede de forma rápida, precisa e confiável. Com ele, a resolução de problemas, como tentativas de intrusão ou ataques massivos de sobrecarga de rede, são detectados rapidamente, tornando possível a sua prevenção. Além disso, o monitoramento provê informações essenciais para a expansão e controle da rede, prevenindo possíveis problemas de sobrecarga e esgotamento de recursos.

Igualmente importante ao monitoramento de rede é a forma de visualização dos dados obtidos. Grandes quantias de dados quantitativos são de difícil interpretação para o olho humano, que é mais sensível a imagens e figuras (GRÉGIO et al., 2009). Então, a utilização de ferramentas para agrupar e exibir estes dados de forma gráfica agiliza a interpretação do estado da rede, diminuindo o tempo de resposta a ataques e facilitando a identificação de padrões em seu comportamento.

Neste trabalho é apresentada uma solução Web para visualização de dados de rede. O Capítulo 2 contém o embasamento teórico para o trabalho, incluindo noções sobre monitoramento de rede, visualização de dados e trabalhos relacionados. O detalhamento do desenvolvimento é encontrado no Capítulo 3, que apresenta a arquitetura do sistema e as tecnologias utilizadas no processo.

2 BASE TEÓRICA

Este capítulo apresenta o embasamento do presente trabalho, começando com uma introdução sobre monitoramento de rede, em seguida mostrando conceitos sobre análise de rede e visualização de dados, e finalizando com trabalhos relacionados.

2.1 MONITORAMENTO DE REDE

O monitoramento é crucial ao gerenciamento de redes (CHOWDHURY et al., 2014). Dados precisos sobre o funcionamento de uma rede são essenciais para tarefas de monitoramento, como engenharia de tráfego e detecção de instrusão. Ao mesmo tempo, ferramentas de monitoramento precisam gerar dados sobre a rede em tempo real sem prejudicar seu funcionamento.

O presente trabalho consiste no desenvolvimento de um sistema Web de visualização de dados de monitoramento de rede. O agrupamento dos dados utilizados pelo sistema é feitos por alguns trabalhos relacionados, tendo como principal “Um sistema por processamento de fluxos aplicado à análise e monitoramento da rede” (HAAS et al., 2019), que consiste em um sistema cujo objetivo é fazer uma análise de rede e alimentar uma base de dados com seu resultado. Ele é estruturado para servir redes de grande porte e, por este motivo, é baseado na arquitetura Lambda, uma arquitetura abordada por (MARZ; WARREN, 2015) para processamento em tempo real de *Big Data*.

Os dados obtidos pelo trabalho de Haas et al têm como principal objetivo a identificação de anomalias, tais como Distributed Denial of Service (DDoS), ataques de força bruta e varredura de portas sobre os protocolos Hypertext Transfer Protocol (HTTP), Secure Shell (SSH) e Transmission Control Protocol (TCP) e, a partir dos resultados, obter informações de organização e país sobre o Internet Protocol (IP), responsáveis por estas práticas.

2.2 ARQUITETURA LAMBDA

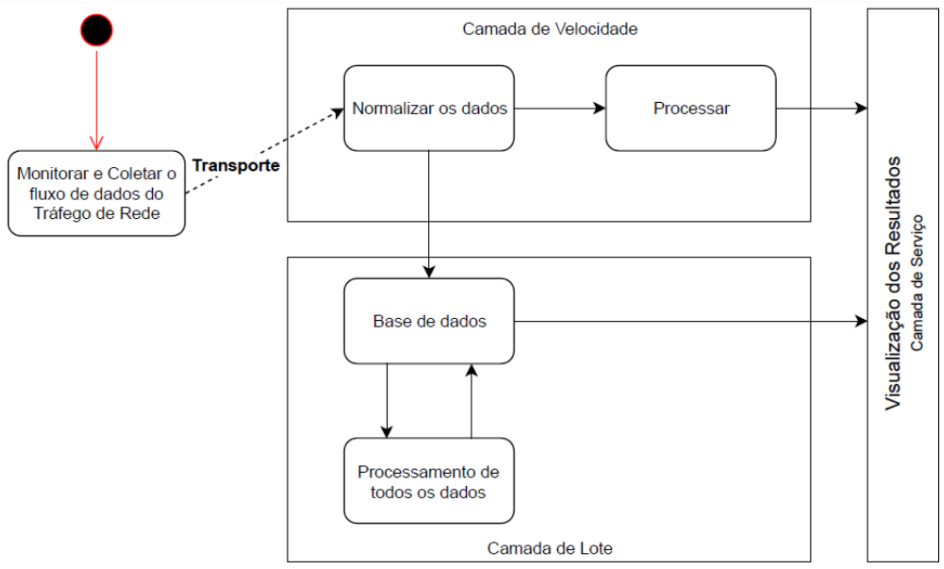
A Arquitetura Lambda foi criada para resolver o problema de processamento arbitrário de grandes quantias de dados em tempo real. Para isso, ela divide o processamento em três camadas:

- Lote: contêm uma grande base de dados centralizada de crescimento constante. Funciona processando os dados contidos nesta base e os disponibilizando em *views*;

- Velocidade: camada responsável pelo tempo real. Trabalha apenas com dados novos, os processando e disponibilizando diretamente em *views* de velocidade;
- Serviço: cria uma visão dos dados processados pelas camadas anteriores.

No sistema de Haas, os dados de rede são obtidos por um script de monitoramento e coleta de fluxo de dados de Tráfego de Rede. Após isso, são transportados e disponibilizados para processamento, e, em seguida, normalizados pela camada de velocidade, que adiciona informações e filtra os dados de acordo com a característica de sua conexão. A partir disso, os dados podem ou ser processados e encaminhados para a camada de serviço, ou armazenados para serem utilizados na camada de lote, que, por sua vez, os analisa, agrupa e classifica, salvando o resultado deste processo em uma base de dados. A camada de serviço, então, pode consultar esta base para visualizar as informações processadas. A Figura 2.1 mostra o modelo da arquitetura do sistema desenvolvido por Haas.

Figura 2.1 – Modelo da arquitetura do sistema de monitoramento de rede proposto por Haas.



Fonte: (HAAS et al., 2019)

O presente trabalho é inserido na camada de serviço desta arquitetura. Ele é responsável por automatizar o processo de agrupamento e filtro dos dados e exibi-los de forma que estes sejam de fácil interpretação.

2.3 VISUALIZAÇÃO DE DADOS

Visualização de dados pode ser explicada como uma forma de obter conhecimento a partir de um conjunto de informações. Segundo (TELEA, 2014), a visualização visa:

- Prover respostas à dúvidas concretas a respeito de determinado problema;
- Prover fatos sobre algum problema não mapeado.

Para Edward R. Tufte et al, "*... de todas as formas de analisar e comunicar informação estatística, gráficos bem feitos sobre dados são geralmente ao mesmo tempo a mais simples e mais poderosa*" (TUFTE, 2001). Figuras, imagens e gráficos sobre dados simplificam sua visualização e, consequentemente, facilitam o processo de obtenção de conhecimento através deles (FUNKHOUSER, 1937).

Tendo como base redes de grande porte, existe uma imensa quantidade de dados que necessitam ser analisados de modo a prover alguma informação que possa ajudar na análise da rede, seja na identificação de uma anomalia ou na obtenção de algum conhecimento importante para sua expansão. Ao mesmo tempo, segundo (GRÉGIO et al., 2009), a análise de dados deste tipo "*...costuma ser impraticável para analistas humanos, bem como pode não gerar resultados apresentáveis do ponto de vista de inteligência dependendo das ferramentas que forem utilizadas para extração de informações.*".

Engenheiros de rede precisam utilizar ferramentas que integram formas convencionais de monitoramento de rede com ferramentas inovadoras de exibição de dados, que forneçam para o usuário final uma maneira fácil de entender o estado da rede (TEN et al., 2009). Gráficos, figuras e mapas são alguns exemplos de ferramentas que podem ser utilizadas para a melhor visualização destes dados. Aliado a isso, o uso de computadores para preparar, organizar e visualizar dados possibilita a exploração de dados de formas novas, eficientes e interessantes (GRÉGIO et al., 2009).

2.4 TRABALHOS RELACIONADOS

Existem diversos trabalhos nas comunidades de Código Aberto e Fechado, que se concentram em desenvolver soluções para realizar a análise e visualização dos dados de rede. Esta seção apresenta uma breve discussão sobre os trabalhos relacionados, considerando que as abordagens com maior relevância para esse estudo serão descritas de forma detalhada.

Na comunidade de Código Aberto existem dois principais sistemas focados na exibição dos dados em uma interface web, o *NfSen* (SWITCH, 2007) e o *Stager* (OSLEBO, 2006). Ambos apresentam dados de rede já capturados e processados. Para a análise, os

dois sistemas utilizam gráficos de linha para mostrar dados temporais agregados. Para a base de dados, ambos utilizam sistemas de gerenciamento eficientes para garantir busca e inserção de dados de forma ágil.

Outros sistemas que tratam da visualização de dados de rede em interfaces web são o *Nagios* (ENTERPRISES, 2015), de código fechado, e o *ntop* (DERI; SUIN, 1999), de código aberto, possuindo também versões pagas. *Nagios* traz o monitoramento de rede com um de seus produtos, o *Network Monitoring*, que inclui o processo de análise de rede, processamento dos dados e exibição deles em um *dashboard*. O *ntop*, assim como o *Nagios*, possui serviços para análise, processamento e exibição dos dados. Ele organiza as exibições principalmente por porta, localidade e quantidade de tráfego, permitindo filtro por período (selecionando data inicial e data final). Ambos, *Nagios* e *ntop*, utilizam formas altamente gráficas para exibição dos dados e organizam seus dados de forma temporal.

Voltado para o lado de busca e tratamento de dados, o Elasticsearch (B.V., 2019) é um sistema de código aberto para busca e análise de dados. Ele funciona como se fosse uma base de dados, indexando os dados de maneira que possibilite seu processamento em tempo real (near-real-time). Porém, ele não possui uma camada de serviço, que pode integrar de dados na camada de velocidade e lote da arquitetura Lambda, além de não possuir por padrão uma forma de visualização dos dados.

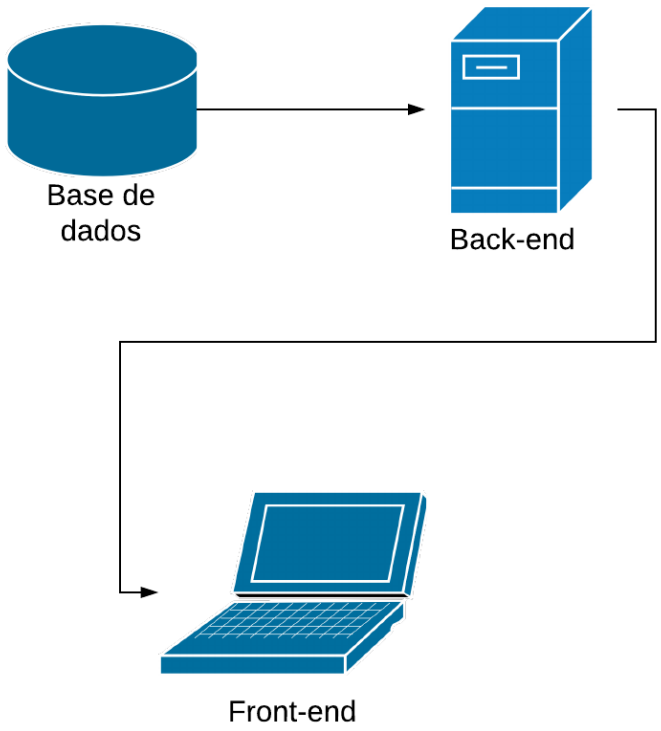
3 DESENVOLVIMENTO

Neste capítulo, as etapas do desenvolvimento deste trabalho são detalhadas. Inicialmente, sua arquitetura é detalhada. Em seguida, são descritos seu front-end e back-end e, finalmente, são explicadas as tecnologias utilizadas para seu desenvolvimento.

3.1 ARQUITETURA

A arquitetura do sistema é dividida em três partes: uma base de dados, uma aplicação back-end e uma interface front-end. A base de dados contém dados atuais e históricos sobre o estado da rede. O back-end faz consultas nesta base, formatando e agrupando os dados obtidos para enviar à interface front-end, que os exibe de forma gráfica. A Figura 3.1 apresenta um modelo desta arquitetura.

Figura 3.1 – Modelo da arquitetura do sistema.



Fonte: Criado pelo autor

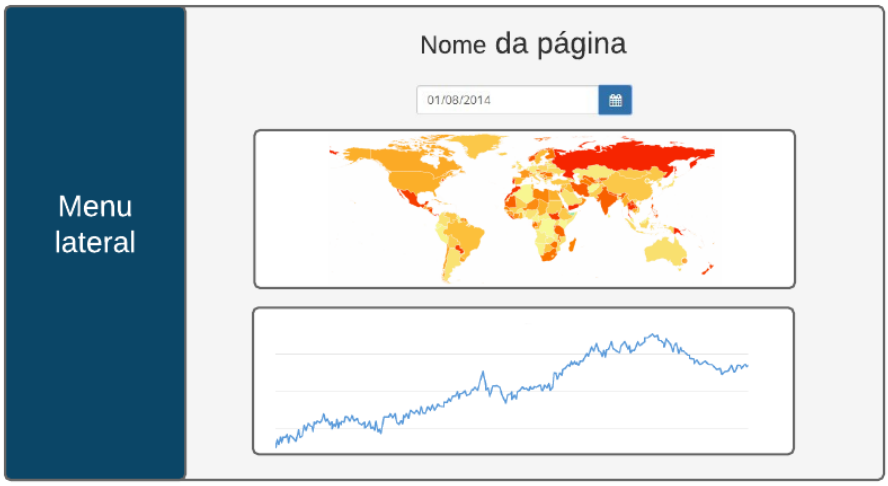
3.2 FRONTEND

Esta seção traz detalhamento sobre o Frontend do sistema. Começa detalhando o modelo padrão das páginas, logo depois mostrando seu fluxograma, detalhando o conteúdo de cada página. Finalmente, descreve seu conteúdo.

3.2.1 Modelo de páginas

As páginas foram organizadas em um modelo padrão, composto por um menu vertical posicionado na lateral esquerda da página e o conteúdo da página, ocupando o restante do espaço. Em todas as páginas, com exceção da página inicial, existe também um campo para seleção de período para filtragem dos dados. A Figura 3.2 demonstra o modelo seguido no desenvolvimento das páginas do sistema.

Figura 3.2 – Modelo de página do sistema.



Fonte: Criado pelo autor

Para exibição dos dados, são utilizados gráficos interativos e tabelas. A interatividade destes gráficos consiste na possibilidade de isolar determinado agrupamento da amostra (exibir somente conexões pelo protocolo SSH de uma amostra com SSH, DHCP e HTTP agrupados) e na capacidade de aplicar *zoom*, facilitando a visualização de um agrupamento grande de dados.

Os modelos de gráfico serão padronizados para cada tipo de amostra:

- Dados temporais agrupados por hora: gráficos de linha;
- Dados temporais agrupados por dia: gráficos de barra verticais;

- Dados percentuais: gráficos de pizza;
- Dados agrupados por entidades: gráficos de barra horizontais;
- Dados geográficos: mapa-múndi de calor separado por país, exibindo cada um com uma tonalidade de cor diferente, que é definida pela quantidade de acessos provenientes daquela localização.

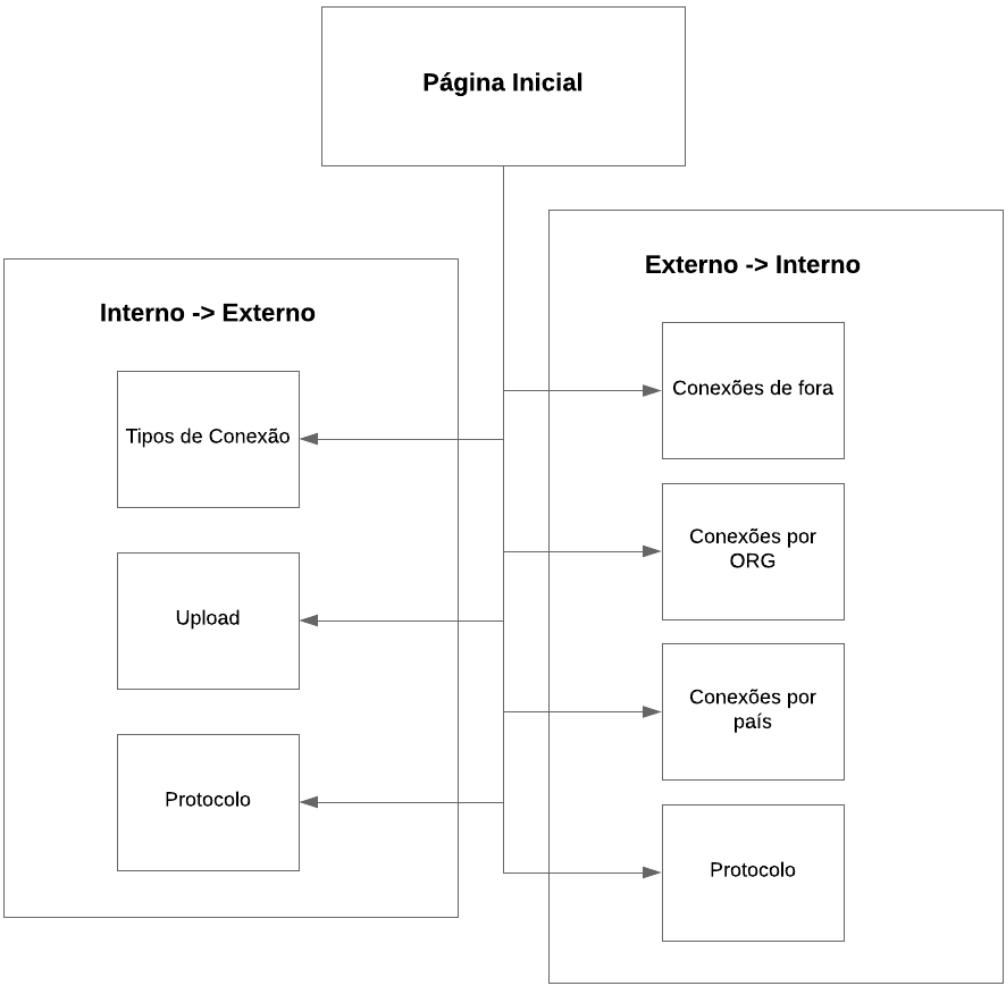
3.2.2 Fluxograma e conteúdo

As páginas foram organizadas em duas seções, que são:

- **Conexões externo-interno**: exibe informações sobre dados de conexões externas acessando informações locais e contém as páginas Conexões de Fora, Conexões por ORG, Conexões por País e Protocolos;
- **Conexões interno-externo**: mostra dados de conexões locais acessando recursos externos e contém as páginas Tipos de conexão, Upload e Protocolos.

O sistema começa carregando a página inicial e pode ser navegado pelo menu lateral, presente em todas as páginas. A Figura 3.3 representa o fluxograma das páginas do sistema, inserindo-as em suas devidas categorias.

Figura 3.3 – Fluxograma de páginas.



Fonte: Criado pelo autor

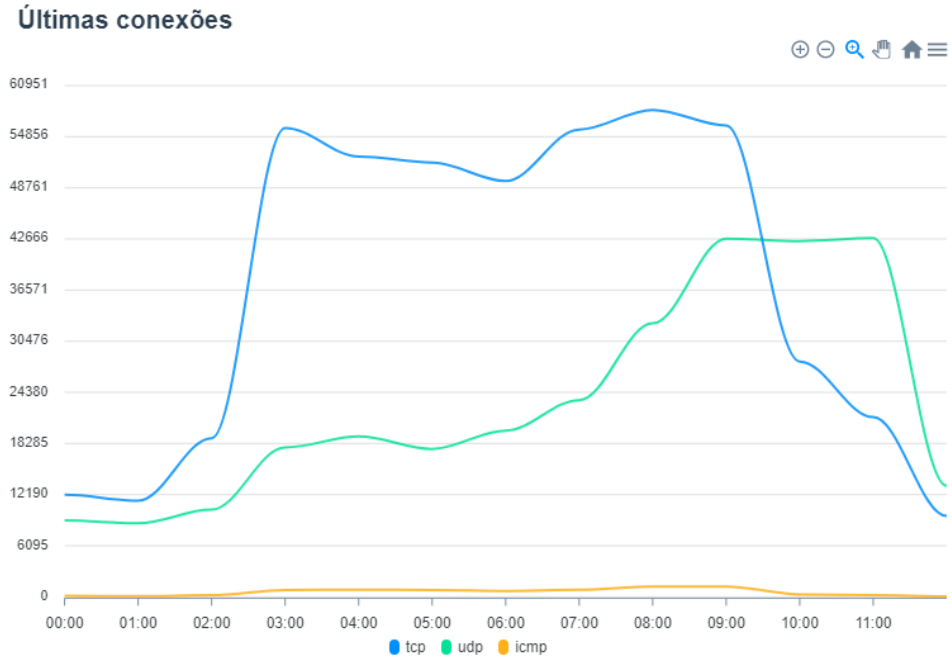
3.2.2.1 Página Inicial

Mostra os dados das conexões recebidas nas últimas 12 horas, apresentando um gráfico temporal de linha agrupado por hora, um gráfico de pizza com os dados totais agrupados e um mapa-múndi de calor, que exibe os dados geográficos do período. Esta é a única página que não contém filtro, considerando que seu objetivo é de representar uma ideia geral do estado da rede.

Como citado acima, os dados agrupados por hora desta página são exibidos em um gráfico de linha. O gráfico gerado é representado pela Figura 3.4 e apresenta elementos interativos. Na parte superior direita da figura é apresentado um menu com opções de controle de zoom e a opção de salvar gráfico gerado como uma imagem nos formatos

PNG e SVG.

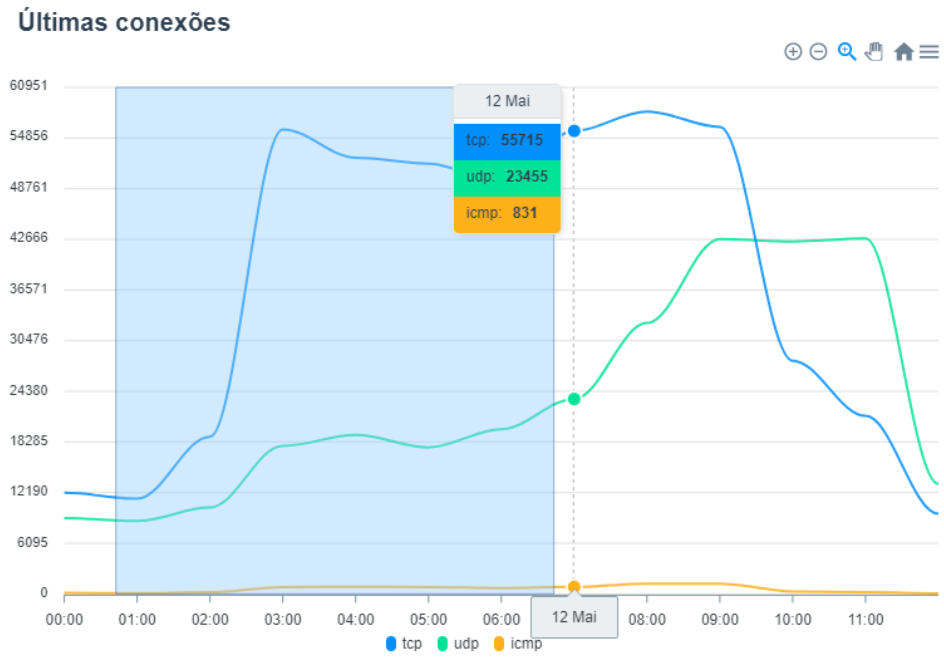
Figura 3.4 – Gráfico de linha temporal exibido na página inicial.



Fonte: Criado pelo autor

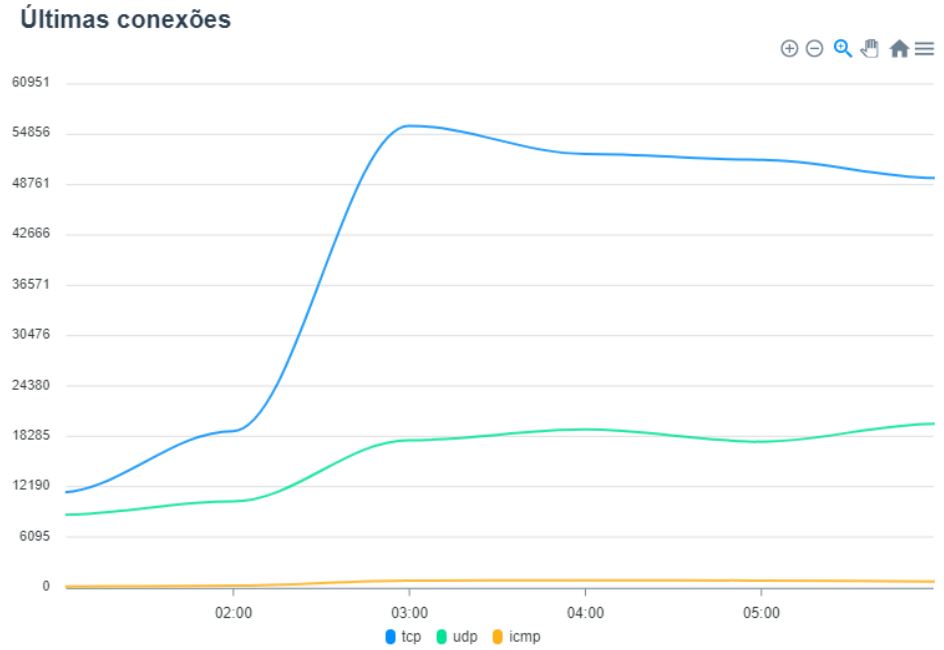
O gráfico permite o foco de determinado período (*zoom*). Ao selecionar um intervalo de tempo com o cursor do mouse, o gráfico é alterado para exibir somente o período selecionado, permitindo voltar para o estado inicial ao clicar o botão com o ícone de uma casa no menu do gráfico. O processo de aplicação de *zoom* e seu resultado é demonstrado nas figuras 3.5 e 3.6.

Figura 3.5 – Seleção de zoom no gráfico de linha gerado.



Fonte: Criado pelo autor

Figura 3.6 – Aplicação de zoom no gráfico de linha gerado.

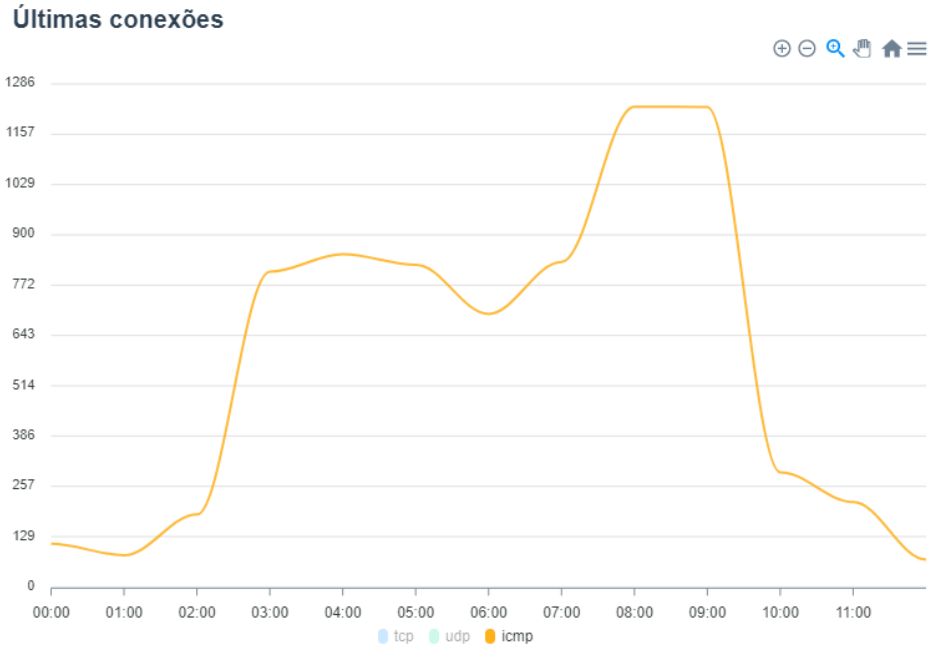


Fonte: Criado pelo autor

Além disso, o gráfico também permite o isolamento de amostras. Para exemplificar este processo, considerando o gráfico demonstrado na Figura 3.4, é possível remover qualquer uma das amostras dele (tcp, udp ou icmp) clicando na sua respectiva legenda

que está posicionada na parte central inferior do gráfico. O processo de isolamento é demonstrado na Figura 3.6, onde foi isolada somente a amostra *icmp*.

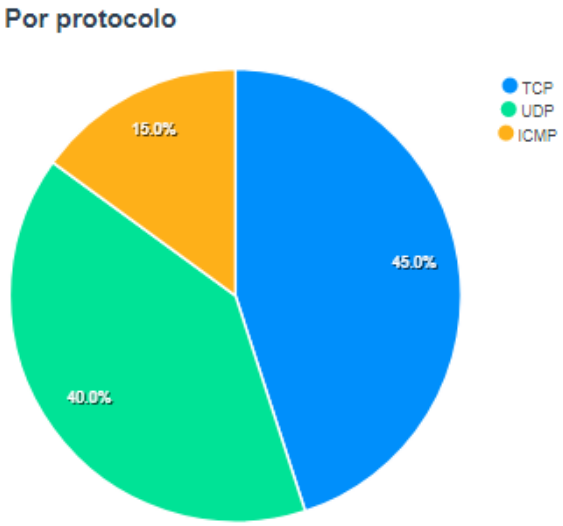
Figura 3.7 – Aplicação do isolamento de amostras no gráfico de linha da Figura 3.4.



Fonte: Criado pelo autor

Nesta página também está inserido um gráfico de pizza com os dados totais das últimas 12 horas. Este foi feito no modelo representado pela figura 3.8.

Figura 3.8 – Gráfico comparativo de pizza presente na Página Inicial.

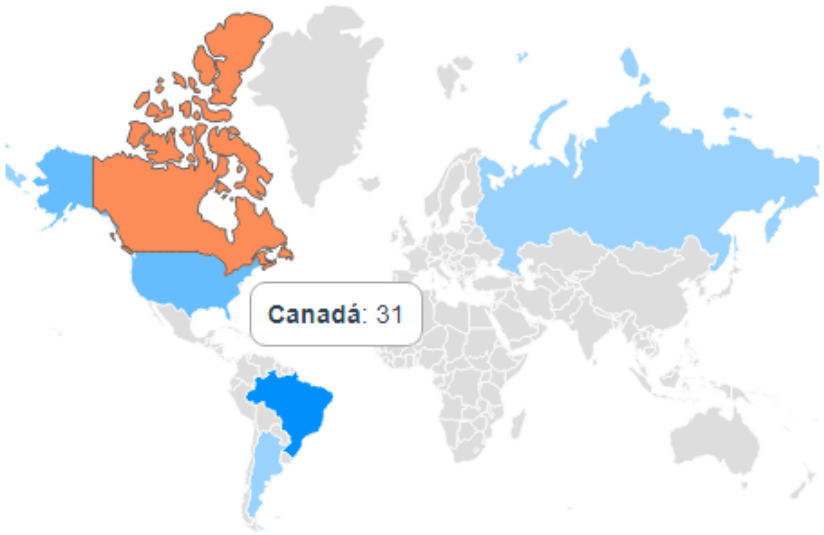


Fonte: Criado pelo autor

Por último, é mostrado um mapa-múndi de calor, que representa os países de onde vieram os acessos do período selecionado. Ao passar o cursor do mouse por algum país

(ação chamada de *hover*) é apresentado em um *tooltip* (pequena janela aberta com destaque em relação aos outros elementos da página) com os dados relacionados a ele, que neste caso são nome e número de acessos. O modelo de mapa, assim como a interatividade com o efeito de *hover* são representados pela Figura 3.9.

Figura 3.9 – Mapa-múndi de calor presente na Página Inicial.



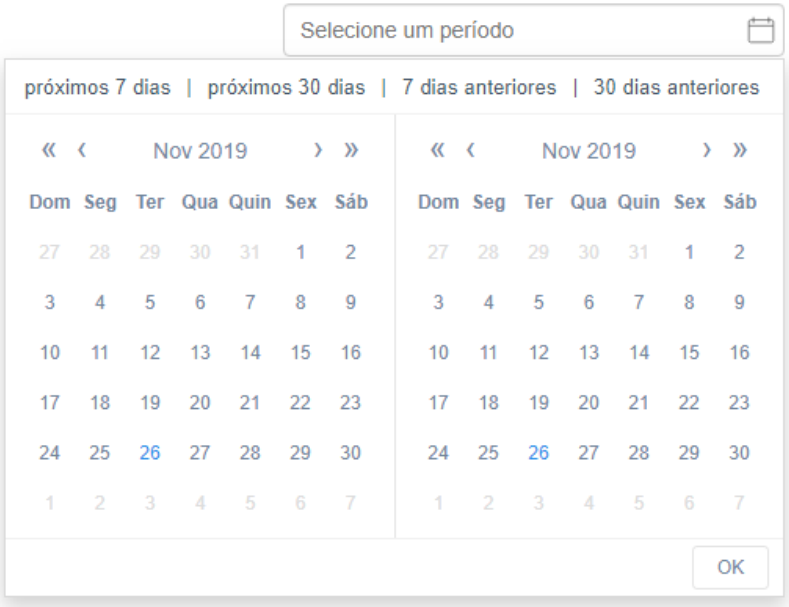
Fonte: Criado pelo autor

3.2.2.2 Páginas Tipos de conexão e Conexões de fora

Estas duas páginas possuem o mesmo *layout*, relacionando o número de conexões (recebidas no caso da página Conexões de fora e feitas no caso da Tipos de conexão) com seu tipo (TCP, UDP, ICMP). Os dados exibidos nelas podem ser filtrados por período e são agrupados por hora, dia e total do período.

Esta, e todas as outras páginas do sistema, exceto a Página Inicial, contém um campo de seleção de período para filtragem dos dados. Este possibilita escolher uma data inicial e final e traz algumas opções pré-definidas configuráveis para esta seleção, como “7 dias anteriores” e “30 dias anteriores”. O campo de filtro está representado na Figura 3.10. Ao clicar no botão “OK”, o sistema automaticamente faz uma busca para a API com o novo filtro e atualiza a página com os dados obtidos.

Figura 3.10 – Filtro por período.

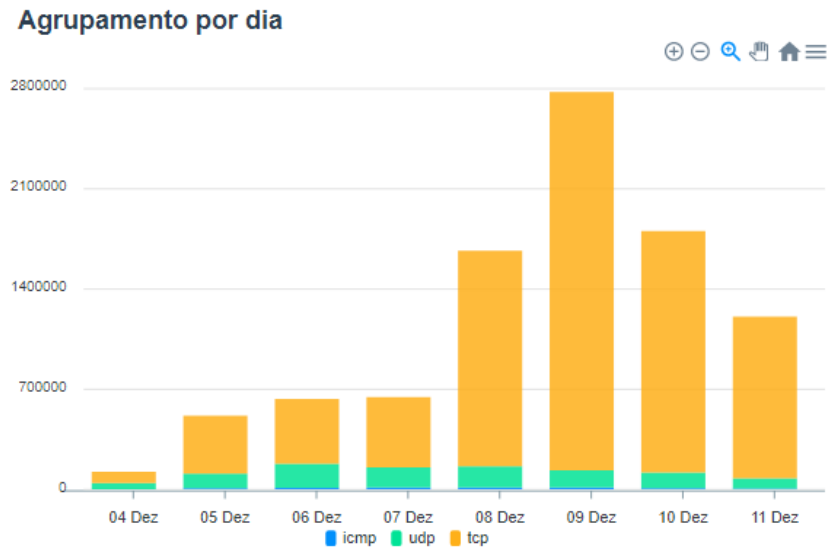


Fonte: Criado pelo autor

Os dados agrupados por hora são exibidos em um gráfico de linha, com o mesmo comportamento do citado na Página Inicial. Consiste em um gráfico interativo que permite visualizar as amostras obtidas no período selecionado, trazendo consigo a possibilidade de isolar amostras e aplicar *zoom* para visualização do comportamento do gráfico com mais precisão.

Um gráfico de barras verticais é utilizado para demonstrar os dados agrupados por dia. Este também é interativo, contendo as funcionalidades já descritas de *zoom* e isolamento de amostras. O gráfico gerado é representado na Figura 3.11.

Figura 3.11 – Modelo de gráfico de barras vertical.



Fonte: Criado pelo autor

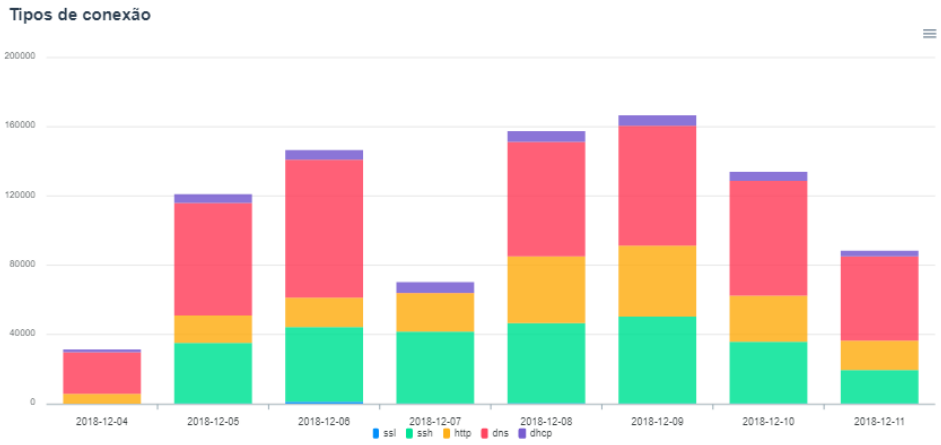
Já os dados totais são exibidos com um gráfico de pizza, semelhante ao descrito na Página Inicial e também em uma tabela posicionada ao lado deste.

3.2.2.3 Páginas Protocolos

No sistema existem duas páginas com o nome de Protocolos, uma na seção Interno-externo e outra na Externo-Interno. A diferença entre elas é que a primeira representa conexões feitas internamente e a segunda representa conexões recebidas. Apesar de possuírem dados diferentes, estas possuem o mesmo layout, que consiste em um filtro de seleção de período, um gráfico de barras verticais, que agrupa os diferentes protocolos por dia, e um gráfico de pizza que, juntamente com uma tabela, mostra os dados totais do período.

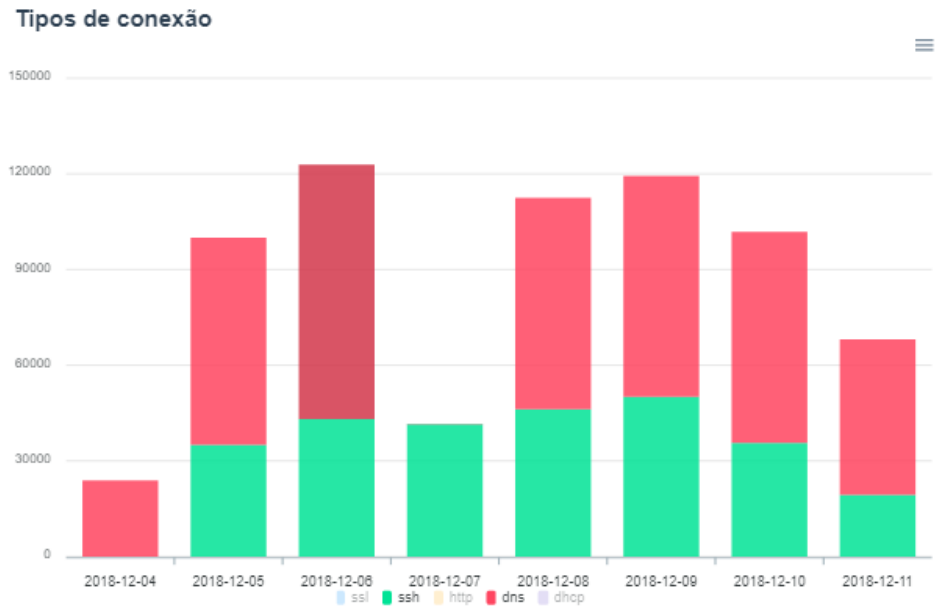
O gráfico de pizza contido é semelhante ao citado na página inicial e o de barras semelhante ao citado nas páginas Tipos de Conexão e Conexões de Fora. O modelo do gráfico de barras gerado, juntamente com uma exemplificação de sua interatividade, é demonstrado nas Figuras 3.12 e 3.13.

Figura 3.12 – Gráfico de barras vertical da página protocolos.



Fonte: Criado pelo autor

Figura 3.13 – Gráfico de barras vertical da página protocolos exibindo somente os dados dos protocolos ssh e dns.

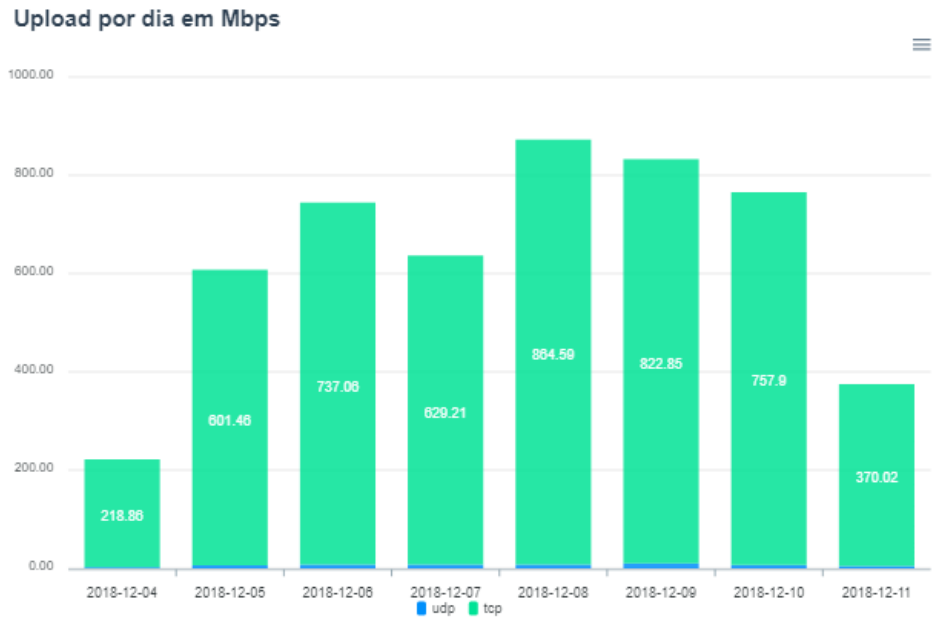


Fonte: Criado pelo autor

3.2.2.4 Página upload

Esta página mostra as informações de download de dados internos por requisições externas. Os dados obtidos são diferenciados pelos protocolos UDP e TCP e ordenados por tempo. Ela contém um campo para seleção de período e um único gráfico de barras verticais, semelhante ao presente nas páginas Tipos de Conexão e Conexões de Fora, que apresenta os dados de upload agrupados por dia. A Figura 3.14 apresenta um modelo do gráfico gerado.

Figura 3.14 – Gráfico de barra vertical da página upload.

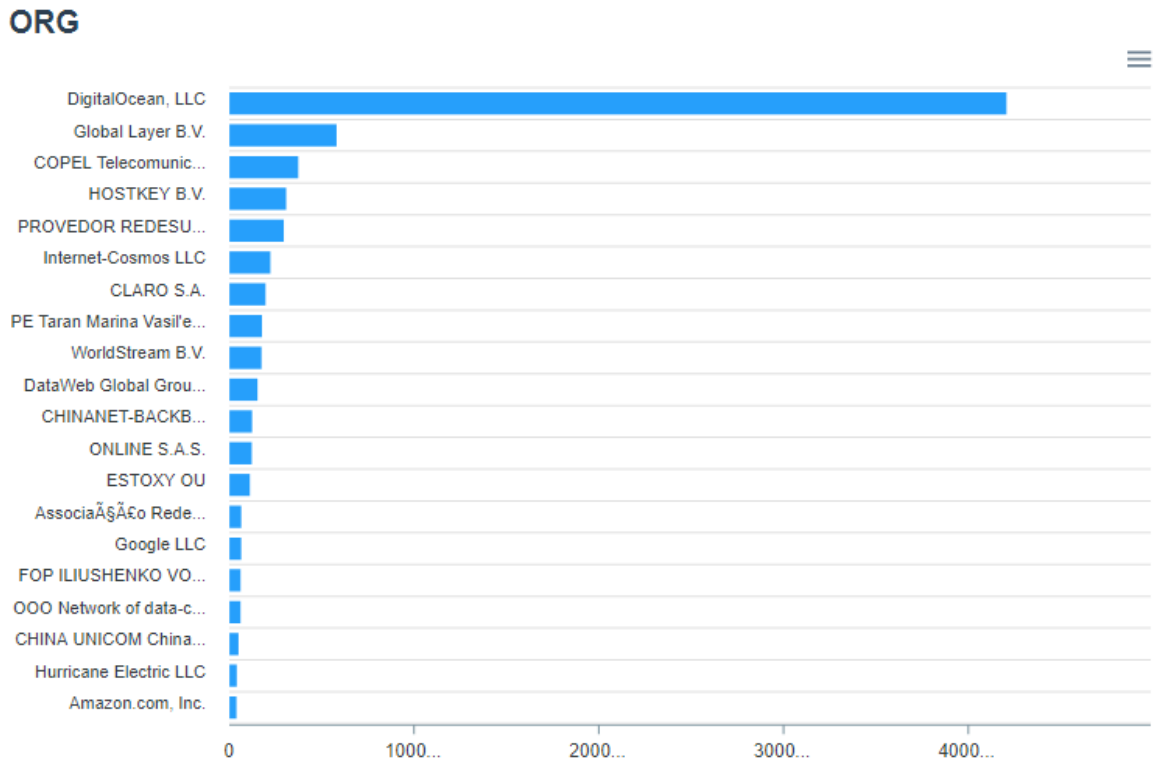


Fonte: Criado pelo autor

3.2.2.5 Página Conexões por ORG

Juntamente com o filtro por período, apresenta somente um gráfico de barras horizontal. Este representa as conexões totais do período selecionado agrupadas pelo nome de sua organização de origem. A Figura 3.15 representa o modelo gerado para este gráfico.

Figura 3.15 – Gráfico de barra horizontal da página Conexões por ORG.



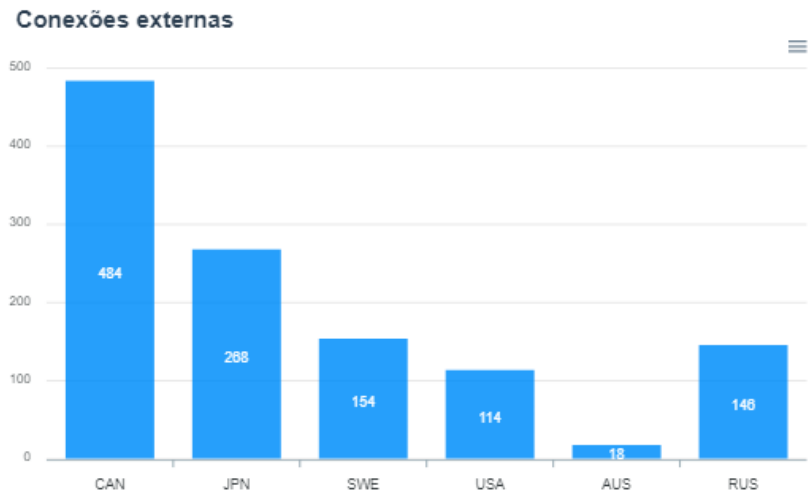
Fonte: Criado pelo autor

3.2.2.6 Página Conexões por país

A página de conexões por país agrupa dados de rede do período selecionado e os ordena pelo país de origem. Ela contém um gráfico de barras verticais que demonstra os dados de conexões externas ao Brasil, um mapa-múndi de calor que representa estes dados de forma dinâmica e uma tabela com os dados totais do período.

O gráfico de barras desta página mostra somente o número de conexões agrupados por país de origem. Nele, os dados relacionados ao Brasil são excluídos já que o número de conexões recebidas por ele é muito maior que o a recebida pelo restante dos países, o que faz com que os dados destes fiquem difíceis de visualizar. O modelo do gráfico gerado é representado pela Figura 3.16.

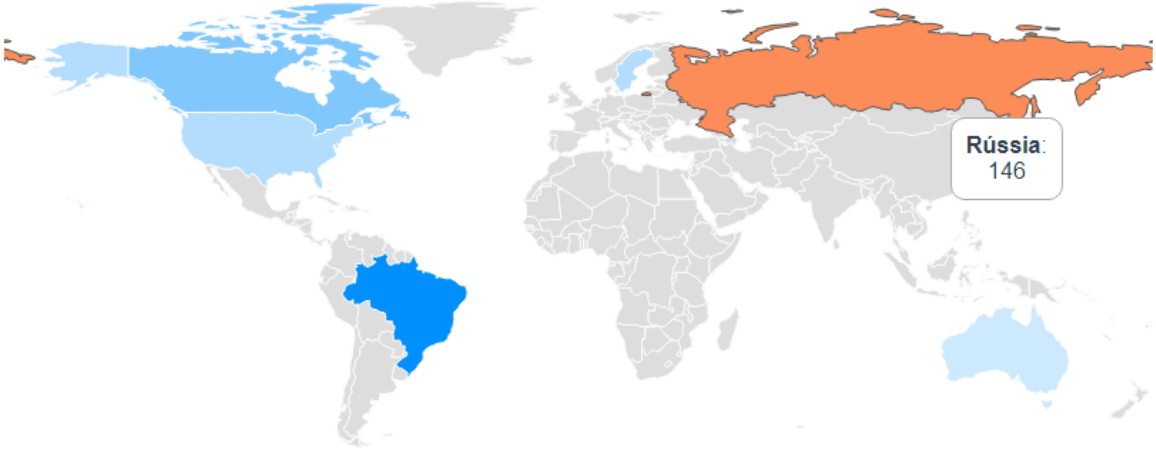
Figura 3.16 – Gráfico de barras vertical da página Conexões por país.



Fonte: Criado pelo autor

O mapa-múndi desta página é semelhante ao gerado na página inicial. Ele também possui os efeitos de *hover* para demonstração dos dados absolutos, com a única diferença sendo o modelo de mapa gerado. Nesta página, como o mapa possui o maior destaque dentre todos os elementos, ele é o maior elemento exibido. A figura 3.17 exibe o mapa-múndi de calor gerado.

Figura 3.17 – Mapa-múndi de calor exibido na página Conexões por País.



Fonte: Criado pelo autor

3.3 BACK-END

O back-end funciona às bases do protocolo HTTP, provendo rotas que podem ser acessadas com o método GET, retornando um objeto com os dados relacionados à rota requisitada ou uma mensagem de erro.

As rotas geradas são subdivididas nas categorias interno e externo e possuem *endpoints* (termo final da rota) específicos para cada tipo de dado requisitado. Por exemplo, existe o endpoint *protocolos* em ambas as categorias, um com a rota *externo/protocolos* e outra com *interno/protocolos*, sendo que as duas retornam diferentes amostras de dados. Todas as rotas possuem a capacidade de receber um filtro temporal por parâmetro, composto de duas datas separadas por hífen. Caso algum *endpoint* seja acessado sem o envio de filtro, este retorna os dados relacionados aos 7 dias anteriores ao dia atual.

Além de fornecer as rotas, o sistema back-end prepara os dados recebidos de maneira que estes sejam mandados para o front-end de forma adequada para seu uso. As informações vindas do banco são preparadas diferentemente para cada *endpoint*, sendo transformadas em um único objeto JavaScript (JSON) (CROCKFORD, 2006) organizado. Os *endpoints* são:

- **Conexões:** Existe em ambos os tipos de conexão (internas e externas). Nele, são requisitados do banco de dados as informações relacionados à conexões por hora do período selecionado. É retornado um objeto contendo todos os dados do período. O sistema os agrupa por tipo de conexão e forma um único objeto a ser enviado para o front-end. Este objeto é composto de várias listas nomeadas, uma para cada tipo de conexão, e uma lista contendo os intervalos de tempo do período. Os dados contidos nas listas de conexões são organizados de forma que seus índices sejam equivalentes ao índice da dos intervalos, ou seja, o dado contido no índice 1 de uma conexão é relacionado ao contido no índice 1 dos intervalos de tempo.

- **Relação:** É contido nos tipos de conexão internas e externas. Busca dados da base e os agrupa em duas listas: uma contendo os tipos de acessos (*icmp*, *dhcp*, *tcp*, ...) e outra contendo o número total de conexões de cada tipo no período requisitado. O relacionamento entre as listas é feito por índice, como mencionado no *endpoint* conexões.

- **Organizações:** Busca e prepara dados relacionados às organizações, estando contido na categoria de conexões externas. O sistema busca dados de um determinado período e retorna duas listas relacionadas por índice, uma contendo os nomes das organizações e outra contendo o número de acessos relacionados a elas.

- **Por dia:** É contido em ambas as categorias de conexões, internas e externas. Busca da base de dados informações relacionadas às conexões do período selecionado

agrupadas por dia. Após isso, as agrupa da mesma forma que no *conexões*, montando uma lista para cada tipo de conexão e uma para o intervalo de tempo a que estas listas estão relacionadas, organizando-as por índice.

- **Upload:** Busca na base dados relacionados à upload agrupados por dia no período selecionado. Retorna duas listas, uma contendo os intervalos de tempo e outra contendo a quantidade de dados enviados, em megabytes. As duas listas são relacionadas por índice.
- **País:** Busca na base os dados totais do períod selecionado, organizados por ÌP de origem. Com isso, o sistema busca o país de origem dos IPs recebidos e monta um objeto, contendo sigla do país e a quantidade de acessos provenientes dele. É o único *endpoint* que não organiza o retorno em listas.

Em primeira instância, o back-end tem por objetivo atuar na camada de lote da arquitetura Lambda. Ou seja, ele se conecta a uma base de dados predefinida e, por meio dela, busca os dados necessários para preparação e envio ao Front-end.

3.4 TECNOLOGIAS UTILIZADAS

Para chegar às funcionalidades finais do front-end foi utilizado o Vue.js (YOU, 2018), framework JavaScript para construção de interfaces de usuário de código aberto. Ele utiliza como base a Hypertext Markup Language 5 (HTML5), Cascading Style Sheets 3 (CSS3) e JavaScript. Possui uma biblioteca principal focada exclusivamente na camada visual e um ecossistema de desenvolvimento que torna fácil a adição de novas bibliotecas e integração com diferentes aplicações.

O back-end foi desenvolvido utilizando o micreframework Flask (RONACHER, 2018), que é escrito em Python e contém um *toolkit* para aplicações com *Web Server Gateway Interface* (WSGI). Este foi construído nas bases da arquitetura REST (Representational State Transfer), que pode ser caracterizada como um tipo de arquitetura para desenvolvimento web, que estabelece características fundamentais e boas práticas para a construção de aplicações web.

A base de dados em que o presente trabalho se conecta é Apache Cassandra, um sistema de banco de dados NoSQL distribuído e altamente escalável (CASSANDRA, 2014). É a base utilizada na camada de lote do trabalho de Haas et al, possuíndo os dados de monitoramento e análise de rede provenientes dele.

Para a execução do sistema foram utilizados dois contêineres Docker (MERKEL, 2014), uma ferramenta de software que fornece uma camada de abstração e automação de virtualização de nível de sistema operacional. Os contêineres criados são integrados

com uma ferramenta provida pelo próprio Docker, o Docker Compose. Este executa e cofigura o ambiente de execução da API e da interface, fazendo com que os dois sistemas sejam instalados com um só comando e sem problemas com dependências externas.

4 CONCLUSÃO

O presente trabalho teve como objetivo a criação de um sistema para visualização de dados de monitoramento de rede e sua inserção na camada de serviço da arquitetura Lambda. Esta é dividida em três camadas principais: lote, velocidade e serviço. A camada velocidade realiza normalização dos dados e, ou os envia diretamente para a camada de serviço, ou os armazena em uma base de dados. A camada de lote processa os dados contidos na base e os deixa preparados para a camada de serviço. Esta, por sua vez, é responsável pela visualização dos dados fornecidos pelas camadas anteriores.

Para visualizar os dados foi desenvolvida uma interface web front-end. Esta foi construída com base em estudos relacionados a visualização de dados e busca representá-los da melhor forma possível. Ela apresenta dados dinâmicos na forma de gráficos interativos, possibilitando a filtragem dos dados visualizados por período e trazendo funcionalidades como *zoom* e a possibilidade de salvar o gráfico amostrado em uma imagem.

Os dados provenientes da camada de lote vêm de uma API, que é conectada a uma base de dados Apache Cassandra. Ela consulta e processa os dados contidos na base, enviando seu resultado para o sistema front-end.

O resultado obtido foi um sistema completo, que atua na camada de serviço da arquitetura Lambda. Este se liga à camada de lote do sistema que realiza o monitoramento de rede e exibe seus dados dinâmicamente, facilitando assim o processo de visualização e obtenção de conhecimento através deles.

4.1 TRABALHOS FUTUROS

O sistema, da forma que está desenvolvido, atua somente na camada de Lote da arquitetura Lambda. Então, para futura implementação, é necessária a integração do back-end com a camada de velocidade. Assim, utilizando o mesmo front-end desenvolvido no presente trabalho, poderá ser possível a visualização de dados de monitoramento de rede em tempo real.

Além disso, devido a não existência de dados relacionados a ataques e intrusões, fica para um futuro desenvolvimento as páginas e rotas relacionados a este tema.

REFERÊNCIAS BIBLIOGRÁFICAS

B.V., E. **Elasticsearch - Open Source Search Analytics**. 2019. Disponível em: <https://www.elastic.co/>.

CASSANDRA, A. Apache cassandra. **Website. Available online at http://planetcassandra. org/what-is-apache-cassandra**, p. 13, 2014.

CHOWDHURY, S. R. et al. Payless: A low cost network monitoring framework for software defined networks. In: IEEE. **2014 IEEE Network Operations and Management Symposium (NOMS)**. [S.l.], 2014. p. 1–9.

CPD-UFSM. **UFSM em Números**. 2019. Disponível em: <https://portal.ufsm.br/ufsm-em-numeros/publico/index.html;jsessionid=05c3013abc3cd76968f6c0c39cbb>.

CROCKFORD, D. The application/json media type for javascript object notation (json). 2006.

DERI, L.; SUIN, S. Ntop: Beyond ping and traceroute. In: SPRINGER. **International Workshop on Distributed Systems: Operations and Management**. [S.l.], 1999. p. 271–283.

ENTERPRISES, N. **Nagios**. [S.l.]: Feb, 2015.

FUNKHOUSER, H. G. Historical development of the graphical representation of statistical data. **Osiris**, The Saint Catherine Press Ltd., v. 3, p. 269–404, 1937.

GOODALL, J. R.; CONTI, G.; MA, K.-L. **Visualization for Computer Security: 5th International Workshop, VizSec 2008, Cambridge, MA, USA, September 15, 2008, Proceedings**. [S.l.]: Springer Science & Business Media, 2008. v. 5210.

GRÉGIO, A. R. A. et al. Técnicas de visualização de dados aplicadas à segurança da informação. **Campinas, São Paulo**, 2009.

HAAS, A. et al. Um sistema por processamento de fluxos aplicado à análise e monitoramento da rede. Universidade Federal de Santa Maria, 2019.

MARZ, N.; WARREN, J. **Big Data: Principles and best practices of scalable real-time data systems**. [S.l.]: New York; Manning Publications Co., 2015.

MERKEL, D. Docker: lightweight linux containers for consistent development and deployment. **Linux Journal**, v. 2014, n. 239, p. 2, 2014.

OSLEBO, A. Stager a web based application for presenting network statistics. In: IEEE. **2006 IEEE/IFIP Network Operations and Management Symposium NOMS 2006**. [S.l.], 2006. p. 1–15.

RONACHER, A. Flask (a python microframework). **Dosegljivo: http://flask. pocoo. org.[Dostopano: 20. 7. 2018]**, 2018.

SWITCH. **Netflow Sensor: A graphical web based front end for the nfdump netflow tools**. 2007. Disponível em: <http://nfsen.sourceforge.net/>.

TELEA, A. C. **Data visualization: principles and practice**. [S.l.]: AK Peters/CRC Press, 2014.

TEN, D. W. H. et al. Study on advanced visualization tools in network monitoring platform. In: IEEE. **2009 Third UKSim European Symposium on Computer Modeling and Simulation**. [S.l.], 2009. p. 445–449.

TUFTE, E. R. **The visual display of quantitative information**. [S.l.]: Graphics press Cheshire, CT, 2001. v. 2.

YOU, E. Vue. js. **Diakses dari https://vuejs. org/, pada tanggal**, v. 17, 2018.